



HAL
open science

Rationality and Escalation in Infinite Extensive Games

Pierre Lescanne

► **To cite this version:**

| Pierre Lescanne. Rationality and Escalation in Infinite Extensive Games. 2011. ensl-00594744v1

HAL Id: ensl-00594744

<https://ens-lyon.hal.science/ensl-00594744v1>

Preprint submitted on 20 May 2011 (v1), last revised 8 Feb 2012 (v4)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Rationality and Escalation in Infinite Extensive Games

Pierre Lescanne¹
Université de Lyon, ENS de Lyon, CNRS (LIP),
46 allée d'Italie, 69364 Lyon, France

¹email: Pierre.Lescanne@ens-lyon.fr, phone: +33472728683, fax: +33472728080

Contents

1	Introduction	5
1.1	Rationality and escalation	5
1.2	Infiniteness	6
1.3	Games	7
1.4	Coinduction	9
1.4.1	Backward coinduction and invariants	10
1.4.2	Backward induction vs backward coinduction	10
1.4.3	Von Neumann and coinduction	11
1.4.4	Proof assistants vs automated theorem provers	11
1.4.5	Induction vs coinduction	12
1.4.6	Acknowledgments	12
2	The concepts through examples	13
2.1	A paradigmatic example: the 0,1 game	13
2.2	Coinduction	14
2.2.1	Histories	14
2.2.2	Lazy binary trees	18
3	Games and strategy profiles	21
3.1	Finite Games	21
3.2	Infinite Games	22
3.3	Infinite Strategy Profiles	23
3.3.1	The predicate “ <i>leads to a leaf</i> ”	24
3.3.2	The predicate “ <i>always leads to a leaf</i> ”	25
3.3.3	The \square modality	25
3.3.4	Bisimilarity	26
3.3.5	Game of a strategy profile	26
3.4	Equilibria	26
3.4.1	Convertibility	26
3.4.2	Nash equilibria	28
3.4.3	Subgame Perfect Equilibria	28
3.5	Escalation	32
3.5.1	Escalation informally	32
3.5.2	Escalation as a formal property	32

4	Case studies	33
4.1	The 0, 1 game	33
4.1.1	Two simple subgame perfect equilibria	33
4.1.2	Cutting the game and extrapolating	34
4.1.3	The 0, 1 game has a rational escalation	34
4.2	The dollar auction	34
4.2.1	Equilibria in the dollar auction	35
4.2.2	Escalation is rational	36
4.3	The infinipede	38
5	Related works	41
6	Postface	43
6.1	Finite resources vs infinite resources	43
6.2	Conclusion	44
A	Two subtle points	45
A.1	Equalities	45
A.2	No utility in infinite runs	45
B	About Coq	47
B.1	Functions in the COQ vernacular	47
B.2	Coinduction in COQ	47
B.3	Excerpts of the COQ development	49
B.3.1	Infinite binary trees	49
B.3.2	Infinite games	49
B.3.3	Infinite strategy profiles	49
B.3.4	Convertibility	50
B.3.5	SGPE	50
B.3.6	Nash equilibrium	51
B.3.7	Dollar Auction	51
B.3.8	ALICE stops always and BOB continues always	51
B.3.9	ALICE continues always and BOB stops always	51
B.3.10	Always give up	51
B.3.11	Infinipede	51
B.3.12	Escalation	52

Chapter 1

Introduction

The aim of this of this book is to study infinite games and to prove formally some properties in this framework. As a consequence, we show that the behavior (the madness) of people which leads to speculative crashes or escalation is can be fully rational. Indeed it proceeds from the statement that resources are infinite. The reasoning is based on the concept of coinduction conceived by computer scientists to model infinite computations and used by economic agents unknowingly. When used consciously, this concept is not as simple as induction and we could paraphrase Newton (Bouchaud, 2008): “*Modeling the madness of people is more difficult than modeling the motion of planets*”.

In this chapter we present the three words of the title, namely rationality, escalation and infiniteness.

1.1 Rationality and escalation

We consider the ability of agents to reason and to conduct their action according to a line of reasoning. We call this *rationality*. It is not clear that agents act always rationally. If an agent acts always following a strict reasoning one says that he (she) is rational. To specify strictly this ability, one associates the agent with a mechanical reasoning device, more specifically a Turing machine or a similar decision mechanism based on abstract computations. One admits that in making a decision the agents chooses the option which is the better, in the sense none other will give better payoff, one says that this option is an equilibrium in the sense of game theory. A well-known game theory situation where rationality of agents is questionable is the so-called *escalation*. This is a situation where there is a sequence of decisions wich can be infinite. If many agents¹ act one after the others in an infinite sequence of decisions and if this sequence leads to situations which are worst and worst for the agents, one speaks of *escalation*.

¹Usually two are enough.

1.2 Infiniteness

It is notorious that there is a wall between finiteness and infiniteness, a fact known to model theorists like Fagin (1993) Ebbinghaus and Flum (1995) and to specialists of functions of real variable. Weierstrass (1872) gave an example of the fact that a finite sum of functions differentiable everywhere is differentiable everywhere whereas an infinite sum is differentiable nowhere. This confusion between finite and infinite is at the origin of the conclusion of the irrationality of the escalation founded on the belief that a property of a infinite mathematical object can be extrapolated from a similar property of finite approximations.² As Fagin (1993) recalls, “Most of the classical theorems of logic [for infinite structures] fail for finite structures” (see Ebbinghaus and Flum (1995) for a full development of the finite model theory). The reciprocal holds obviously: “Most of the results which hold for finite structures, fail for infinite structures”. This has been beautifully evidenced in mathematics, when Weierstrass (1872) has exhibited his function:

$$f(x) = \sum_{n=0}^{\infty} b^n \cos(a^n x \pi).$$

Every finite sum is differentiable and the limit, i.e., the *infinite* sum, is not. To give another picture, infinite games are to finite games what fractal curves are to smooth curves (Edgar, 2008). In game theory the error done by the ninetieth century mathematicians³ would lead to the same issue. With what we are concerned, a result which holds on finite games does not hold necessarily on infinite games and vice-versa. More specifically equilibria on finite games are not preserved at the limit on infinite games whereas new types of equilibria emerge on the infinite game not present in the approximation (see the 0, 1 game in Section 2.1) and Section 4.1. In particular, we cannot conclude that, whereas the only rational attitude in finite dollar auction would be to stop immediately, it is irrational to escalate in the case of an infinite auction. We have to keep in mind that in the case of escalation, the game is infinite, therefore reasoning made for finite objects are inappropriate and tools specifically conceived for infinite objects should be adopted. Like Weierstrass’ discovery led to the development of function series, logicians have devised methods for correct deductions on infinite structures. The right framework for reasoning logically on infinite mathematical objects is called *coinduction*.

The inadequate reasoning on infinite games is as follows: people study finite approximation of infinite games as infinite games truncated at a finite location. If they obtain the same result on all the approximations, they extrapolate the result to the infinite game as if the limit would have the same property. But this says nothing since the *infiniteness is not the limit of finiteness*. Instead of reexamining their reasoning or considering carefully the hypotheses their

²In the postface (Chapter 6) we give another explanation: agents stipulate an a priori hypothesis that resources are finite and that therefore escalation is impossible.

³Weierstrass quotes some careless mathematicians, namely Cauchy, Dirichlet and Gauss, whereas Riemann was conscient of the problem.

reasoning is based upon (is the set of resource infinite?) they conclude that humans are irrational. If there is an escalation, then the game is infinite, then *the reasoning must be specific to infinite games, that is based on coinduction*. This is only on this basis that one can conclude that humans are rational or irrational. In no case, a property on the infinite game generated by escalation can be extrapolated from the same property on finite games.

In this book we address these issues. The games we consider may have arbitrary long histories as well as infinite histories. In our games there are two choices at each node⁴, this will not lose generality, since we can simulate finitely branching games in this framework. By König’s lemma, finitely branching, specifically binary, infinite games have at least an infinite history. We are taking the problem of defining formally infinite games, infinite strategy profiles, and infinite histories extremely seriously. By “seriously” we mean that we prepare the land for precise, correct and rigorous reasoning. For instance, an important issue which is not considered in the literature is how the utilities associated with an infinite history are computed. To be formal and rigorous, we expect some kinds of recursive definitions, more precisely co-recursive definitions, but then comes the questions of what the payoff associated with an infinite strategy profile is and whether such a payoff exists (see Section A.2).

1.3 Games

Finite extensive games are represented by finite trees and are analyzed through induction. For instance, in finite extensive games, a concept like *subgame perfect equilibrium* is defined inductively and receives appropriately the name of *backward induction*. Similarly *convertibility* (an agent changes choices in his strategy) has also an inductive definition and this concept is a key for this of *Nash equilibrium*. But *induction*, which has been designed for finitely based objects, no more works on infinite⁵ games, i.e., games underlying infinite trees. Logicians have proposed a tool, which they call *coinduction*, to reason on infinite objects. In short, since objects are infinite and their construction cannot be analyzed, coinduction “observes” them, that is looks at how they “react” to operations (see Section 2.2 for more explanation). In this book, we formalize with coinduction, the concept of infinite game, of infinite strategy profile, of equilibrium in infinite games, of utility (payoff), and of subgame. We verify on the proof assistant COQ that everything works smoothly and yields interesting consequences. Thanks to coinduction, examples of apparently paradoxical human behavior are explained logically, demonstrating a rational behavior.

Finite extensive games have been introduced by Kuhn (1953). But many interesting extensive games are infinite and therefore the theory of infinite extensive games play an important role in game theory, with examples like the *dollar auction game* (Shubik, 1971; Colman, 1999; Gintis, 2000; Osborne and Rubin-

⁴Those choices are often to *stop* or to *continue*.

⁵In this book, *infinite* means infinite and discrete. For us, an infinite extensive game is discrete and has infinitely many nodes.

stein, 1994), the generalized *centipede game*⁶ or infinipede or the 0,1 *game*. From a formal point of view, the concepts associated with infinite extensive games are not appropriately treated in papers and books. In particular, there is no clear notion of Nash equilibrium in infinite extensive game and the gap between finiteness and infiniteness is not correctly understood. For instance in one of the textbooks on game theory, one finds the following definition of *games with finite horizon*: *If the length of the longest derivation is [...] finite, we say that the game has a finite horizon. Even a game with a finite horizon may have infinitely many terminal histories, because some player has infinitely many actions after some history.* Notice that in an infinite game with infinite branching it is not always the case that a longest derivation exists. If a game has only finite histories, but has infinitely many such finite histories of increasing length, there is no longest history. Before giving a formal definition later in the book, let us say intuitively why this definition is inconsistent. Roughly speaking, a history is a path in the ordered tree which underlies the game. A counterexample is precisely when the tree is infinitely branching i.e., when “some player has infinitely many actions”.⁷

Escalation takes place in specific sequential games in which players continue although their payoff decreases on the whole. The *dollar auction* game has been presented by Shubik (1971) as the paradigm of escalation. He noted that, even though their cost (the opposite of the payoff) basically increases, players may keep bidding. This attitude was considered as inadequate and when talking about escalation, Shubik (1971) says this is a paradox, O’Neill (1986) and Leininger (1989) consider the bidders as irrational, Gintis (2000) speaks of *illogic conflict of escalation* and Colman (1999) calls it *Macbeth effect* after Shakespeare’s play. Rebutting these authors, we prove in this book, using a reasoning conceived for infinite structures that escalation is logic and that agents are rational, therefore this is not a paradox and we are led to assert that Macbeth is in some way rational.

This escalation phenomenon occurs in infinite sequential games and only there. Therefore it must be studied in infinite games with adequate tools, i.e., in a framework designed for mathematical infinite objects. Like Shubik (1971) we will limit ourselves to two players only. In auctions, this consists in the two players bidding forever. This statement of rationality is based on the largely accepted assumption that a player is rational if he adopts a strategy which corresponds to a *subgame perfect equilibrium*. To characterize this equilibrium the above cited authors consider a finite restriction of the game for which they compute the subgame perfect equilibrium by *backward induction*⁸. Then they extrapolate the result obtained on the amputated games to the infinite game. To justify their practice, they add a new hypothesis on the amount of money the bidders are ready to pay, called the *limited bankroll*. By enforcing the

⁶Here “generalized” means that the game has an infinite “backbone”.

⁷In chapter 3.3, we define a predicate *leads to leaf* which we think to characterize properly the concept of *finite horizon* which is a property of strategy profiles.

⁸What is called “backward induction” in game theory is roughly what is called “induction” in logic.

finiteness of the game, they exclude clearly escalation. In the amputated game dollar auction, they conclude that there is a unique subgame perfect equilibrium. This consists in both agents giving up immediately, not starting the auction and adopting the same choice at each step. In our formalization in infinite games, we show that extending that case up to infinity is not a subgame perfect equilibrium and we found two subgame perfect equilibria, namely the cases when one agent continues at each step and the other leaves at each step. Those equilibria which correspond to rational attitudes account for the phenomenon of escalation. Actually this discrepancy between equilibrium in amputated games extrapolated to infinite extensions and infinite games occurs in a much simpler game than the dollar auction namely the 0, 1 game which will be studied in this book.

1.4 Coinduction

Like induction, coinduction is based on a fixpoint, but whereas induction is based on the least fixpoint, coinduction is based on the greatest fixpoint, for an ordering we are not going to describe here as it would go beyond the scope of this book. Attached to induction is the concept of inductive definition, which characterizes objects like finite lists, finite trees, finite games, finite strategy profiles, etc. Similarly attached to coinduction is the concept of coinductive definition which characterizes streams (infinite lists), infinite trees, infinite games, infinite strategy profiles etc. An inductive definition yields the least set that satisfies the definition and a coinductive definition yields the greatest set that satisfies the definition. Associated with these definitions we have inference principles. For induction there is the famous *induction principle* used in backward induction. On coinductively defined sets of objects there is a principle like induction principle which uses the fact that the set satisfies the definition (proofs by case or by pattern) and that it is the largest set with this property. Since coinductive definitions allow us building infinite objects, one can imagine constructing a specific category of objects with “loops”, like the infinite word $(abc)^\omega$ (i.e., $abcabcabc\dots$) which is made by repeating the sequence abc infinitely many times.⁹ Other examples with trees are given in Section 2.2.2, with infinite games and strategy profiles in Chapter 3. Such an object is a fixpoint, this means that it contains an object like itself. For instance $(abc)^\omega = abc(abc)^\omega$ contains itself. We say that such an object is defined as a cofixpoint. To prove a property P on a cofixpoint $\sigma = f(\sigma)$, one assumes P holds on σ (the σ in $f(\sigma)$), considered as a sub-object of σ . If one can prove P on the whole object (on $f(\sigma)$), then one has proved that P holds on σ . This is called the *coinduction principle* a concept which comes from Park (1981), Milner and Tofte (1991), and Aczel (1988) and was introduced in the framework we are considering by Coquand (1993). Sangiorgi (2009) gives a good survey with a complete historical account. To be sure to not be entangled, it is advisable to use a proof assistant which implements coinduction, to build and to check the proof. Indeed reasoning with coinduction is sometimes

⁹The notation α^ω for an infinite repetition of the word α is classical.

so counter-intuitive that the use of a proof assistant is not only advisable but compulsory. For instance, we were, at first, convinced that in the dollar auction the strategy profile consisting in both agents stopping at every step was a Nash equilibrium, like in the finite case, and only failing in proving it mechanically convinced us of the contrary and we were able to prove the opposite, namely that the strategy profile “stopping at every step” is not a Nash equilibrium. In the examples of Chapter 4, we have checked every statement using COQ and in what follows a sentence like “we have proved that ...” means that we have succeeded in building a formal proof in COQ.

1.4.1 Backward coinduction as a method for proving invariants

In infinite strategy profiles, the coinduction principles can be seen as follows: a property which holds on a strategy profile of an infinite extensive game is an *invariant*, i.e., a property which is always true, along the histories and to prove that this is an invariant one proceeds back to the past. Therefore the name *backward coinduction* is appropriate, since it proceeds backward the histories, from future to past.

1.4.2 Backward induction vs backward coinduction

One may wonder the difference between the classical method, we call *backward induction* and the new method we call *backward coinduction*. The main difference is that backward induction starts the reasoning from the leaves, works only on finite games and does not work on infinite games (or on finite strategy profiles), because it requires a well-foundedness to work properly, whereas *backward coinduction* works on infinite games (or on infinite strategy profiles). Coinduction is unavoidable on infinite games, since the methods that consists in “cutting the tail” and extrapolating the result from finite games or finite strategies profile to infinite games or infinite strategy profiles cannot solve the problem or even approximate it. It is indeed the same erroneous reasoning as this of the predecessors of Weierstrass who concluded that since:

$$\forall p \in \mathbb{N}, f_p(x) = \sum_{n=0}^p b^n \cos(a^n x \pi),$$

is differentiable everywhere then

$$f(x) = \sum_{n=0}^{\infty} b^n \cos(a^n x \pi).$$

is differentiable everywhere whereas $f(x)$ is differentiable nowhere.

Much earlier, during the IVth century BC, the improper use of inductive reasoning allowed Parmenides and Zeno to negate motion and lead to *Zeno’s paradox of Achilles and the tortoise*. This paradox was reported by Aristotle as follows:

“In a race, the quickest runner can never overtake the slowest, since the pursuer must first reach the point whence the pursued started, so that the slower must always hold a lead.”

Aristotle, Physics VI:9, 239b15

In Zeno’s framework, Zeno’s reasoning is correct, because by induction, one can prove that Achilles will never overtake the tortoise. Indeed this applies to the infinite sequence of races described by Aristotle. In each race of the sequence, the pursuer starts from where the pursued started previous the race and the pursuer ends where the pursued started in the current race. By induction one can prove the truth of the statement “Achilles will never overtake the tortoise” only for a sequence of races. In each race “Achilles does not overtake the tortoise”. For the infinite race for which coinduction would be needed, this result does not hold, which means “Achilles overtakes the tortoise”. By the way, experience tells us that Achilles would overtake the tortoise in a real race, hence the apparent paradox.

1.4.3 Von Neumann and coinduction

As one knows, von Neumann (von Neumann, 1928; von Neumann and Morgenstern, 1944) is the creator of game theory, whereas extensive games and equilibrium in non cooperative games are due to Kuhn (1953) and Nash, Jr. (1950). In the spirit of their creators all those games are finite and backward induction is the basic principle for computing subgame perfect equilibria (Selten, 1965). This is not surprising since von Neumann (1925) is also at the origin of the role of well-foundedness in set theory despite he left a door open for a not well-founded membership relation. As explained by Sangiorgi (2009), research on anti-foundation initiated by Mirimanoff (1917) are at the origin of coinduction and were not well known until the work of Aczel (1988).

1.4.4 Proof assistants vs automated theorem provers

COQ is a proof assistant built by *The Coq development team (2007)*, see Bertot and Castéran (2004) for a good introduction and notice that they call it “interactive theorem provers”, which is a strict synonymous. Despite both deal with theorems and their proofs and are mechanized using a computer, proof assistants are not automated theorem provers. In particular, they are much more expressive than automated theorem provers and this is the reason why they are interactive. For instance, there is no automated theorem prover implementing coinduction. Proof assistants are automated only for elementary steps and interactive for the rest. A specificity of a proof assistant is that it builds a mathematical object called a (formal) proof which can be checked independently, copied, stored and exchanged. Following Harrison (2008) and Dowek (2007), we can consider that they are the tools of the mathematicians of the XXIth century. Therefore using a proof assistant is a highly mathematical modern activity.

The mathematical development presented here corresponds to a COQ script¹⁰ which can be found on the following url's:

<http://perso.ens-lyon.fr/pierre.lescanne/COQ/Book/>
<http://perso.ens-lyon.fr/pierre.lescanne/COQ/Book/SCRIPTS/>

1.4.5 Induction vs coinduction

To formalize structured finite objects, like finite games, one uses *induction*, i.e.,

- a definition of basic objects
 - in the case of natural numbers, induction provides an operator 0 to build a natural number out of nothing,
 - in the case of binary trees, a tree with non node, written \bigcirc ,
 - in the case of finite games induction provides an operator $\langle _ \rangle$ to build a game out of nothing using an function that attribute payoffs to agents.

and

- a definition of the way to build new objects
 - in the case of natural numbers, induction provides an operation *successor* to build a natural number from a natural number,
 - in the case of binary trees, induction provides a binary operator which builds a tree with two trees,
 - in the case of finite games induction provides an operator to build a game out of two subgames and a node.

In the case of infinite objects like infinite games, one characterizes infinite objects not by their construction, but by their behavior. This characterization by “observation” is called *coinduction*. Coinduction is associated with the greatest fixpoint. The proof assistant COQ offers a framework for coinductive definitions and reasonings which are keys of our formalization.

1.4.6 Acknowledgments

My research on game theory started during a visit at JAIST invited by René Vestergaard, then was continued by fruitful with Stéphane Le Roux and Franck Delaplace. A decisive step on infinite games and Nash Equilibrium was done by Matthieu Perrinel. I thank all them.

¹⁰A *script* is a list of commands of a proof assistant.

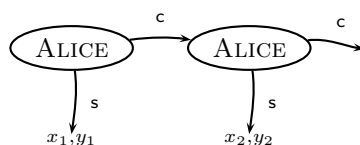
Chapter 2

The concepts through examples

We think that examples are the best way to present concepts. In this chapter we present a simple example of an infinite game useful in what follows and two examples of structures meant to introduce smoothly induction.

2.1 A paradigmatic example: the 0, 1 game

Classically, an extensive game is considered a labelled oriented tree, in which both nodes and arcs are labelled. In other words, there is a set of nodes and a set of arcs. An arc connects a node to another node in such a way that there is no circuit in the graph, i.e., no path which goes from a node to itself when following the arcs. An *internal node* is a node which is connected to another node and an *external node* or a *leaf* is a node which is not connected to another. Internal nodes are labelled by names of players and represents a turn in the game and the label of the internal node tells us which player has the turn. External nodes represent the end of the game and are labelled by the function that assigns a payoff¹ to each player. Arcs are labeled by choices, more precisely choices made by the player who has the turn and show how the choice made by the player who has the turn leads in another position in the game. In our formalization we assume that one can go to a position in which the same player has the turn, like



but this situation never occurs in examples we consider. In this book, we propose

¹or a cost in some cases.

a presentation of game less descriptive and more structural in the line of what is done in computer science when describing infinite computations.

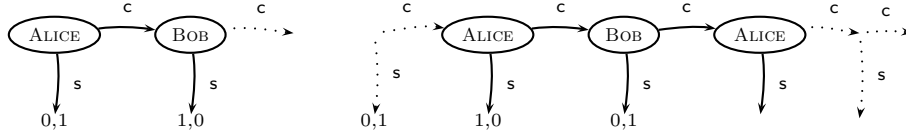


Figure 2.1: The 0,1 game.

To illustrate this description we propose a running example of an infinite extensive game which we call the 0,1 *game* (which write also `zero_one` game) because the only utilities (or payoffs) are 0 and 1. The game is infinite and is represented by a kind of infinite backbone (see figure 2.1) in which each internal node is connected to a leaf and to another internal node. We assume that there are two players, namely ALICE and BOB, hence two labels on the internal nodes. We also assume that players play one after the other. They have two choices, *s* for *stop* and *c* for *continue*. The arc labeled *s* is connected to a leaf and the arc labeled *c* is connected to another internal node. The leaves have labels that are payoff functions. The leaf connected with an internal node labeled with ALICE is labeled with function $\{\text{ALICE} \mapsto 0, \text{BOB} \mapsto 1\}$ (meaning ALICE's payoff is 0 and BOB's payoff is 1) whereas the leaf connected with an internal node labeled with BOB is labeled with function $\{\text{ALICE} \mapsto 1, \text{BOB} \mapsto 0\}$ (meaning the payoffs are reversed). The 0,1 *game* is a specific case of a binary game which are presented using the formalism for defining infinite objects coinductively. Binary games have two kinds of labels called ℓ and r . In the 0,1 game ℓ stands for *c* (continues) and r stands for *s* (stops). In what follows the 0,1 game will be formally defined as a coinductive structure defined by a fixpoint, that is

$$\text{zero_one} = \begin{array}{c} \text{ALICE} \xrightarrow{c} \text{BOB} \xrightarrow{c} \text{zero_one} \\ \downarrow s \qquad \downarrow s \\ 0,1 \qquad \qquad 1,0 \end{array}$$

2.2 Coinduction through examples

We now leave the descriptive approach for the structural approach. We introduce the concept of coinduction through two examples: histories of sequential games, i.e., the sequences of choices performed by agents along the run of a game according to a strategy profile, and binary trees, trees in which there are two subtrees at each node.

2.2.1 Histories

Infinite objects have peculiar behaviors. To start with a simple example, let us have a look at *histories* in games (Osborne, 2004, Chap. 5). In a game, agents make *choices*. In an infinite game, agents can make finitely many choices before

ending, if they reach a terminal node, or infinitely many choices, if they run forever. Choices are recorded in a *history* in both cases. A history is therefore a finite or an infinite list of choices. In this book, we consider that there are two possible choices: ℓ and r (ℓ for “left” and r for “right”). Since a history is a potentially infinite object, it cannot be defined by structural induction.² On the contrary, the type³ *History* has to be defined as a *CoInductive*, i.e., by coinduction, that is a mechanism which defines infinite objects and allows to reason on them. Let us use the symbol $[]$ for the empty history and the binary operator $::$ for non empty histories. When we write $a :: h$ we mean that the history starts with a and follows with the history h . For instance, the finite history $\ell r \ell$ can be written $\ell :: (r :: (\ell :: []))$. If h is the history ℓ^ω (an infinite sequence of ℓ 's) $\ell :: h$ or $\ell :: \ell^\omega$ is the history that starts with ℓ and follows with infinitely many ℓ 's. The reader recognizes that $\ell :: \ell t t^\omega$ is ℓ^ω itself. To define histories coinductively we say the following:

A **coinductive** history (or a finite or infinite history) is

- either the empty history $[]$,
- or a history of the form $a :: h$, where a is a choice and h is a history.

The word “coinductive” says that we are talking about finite or infinite objects. This should not be mixed up with finite histories which will be defined inductively as follows:

An **inductive** history (or a finite history) is built as

- either the empty history $[]$,
- or a finite non empty history which is the composition of a choice a with a finite history h_f to make the finite history $a :: h_f$.

Notice the use of the participial “built”, since in the case of induction, we say how objects are built, because they are built finitely. The 0,1 game has the family of histories $c^*s \cup c^\omega$, meaning that a history is either a sequence of c 's followed by a s , or an infinite sequence of c 's. Consider now an arbitrary infinite binary game⁴ with histories made of ℓ 's and r 's. Let us now consider four families of histories:

\mathcal{H}_0	The family of finite histories
\mathcal{H}_1	The family of finite histories or of histories which end with an infinite sequence of ℓ 's
\mathcal{H}_2	The family of finite histories or infinite histories which contain infinitely many ℓ 's
\mathcal{H}_∞	The family of finite or infinite histories

²In type theory, a type of objects defined by induction is called an *Inductive*, a shorthand we can use for *inductive type*.

³Since we are in type theory, the basic concept is “*type*”. Since we are using only a small part of type theory, it would not hurt to assimilate naive types with naive sets.

⁴An game which does not have a centipede structure, i.e., which does not have a backbone.

We notice that $\mathcal{H}_0 \subset \mathcal{H}_1 \subset \mathcal{H}_2 \subset \mathcal{H}_\infty$. If \mathcal{H} is a set of histories, we write $\ell :: \mathcal{H}$ the set $\{h \in \mathcal{H}_\infty \mid \exists h' \in \mathcal{H}, h = c :: h'\}$. We notice that \mathcal{H}_0 , \mathcal{H}_1 , \mathcal{H}_2 and \mathcal{H}_∞ are solutions of the fixpoint equation :

$$\mathcal{H} = \{[]\} \cup \ell :: \mathcal{H} \cup r :: \mathcal{H}.$$

in other words

$$\begin{aligned} \mathcal{H}_0 &= \{[]\} \cup \ell :: \mathcal{H}_0 \cup r :: \mathcal{H}_0 \\ \mathcal{H}_1 &= \{[]\} \cup \ell :: \mathcal{H}_1 \cup r :: \mathcal{H}_1 \\ \mathcal{H}_2 &= \{[]\} \cup \ell :: \mathcal{H}_2 \cup r :: \mathcal{H}_2 \\ \mathcal{H}_\infty &= \{[]\} \cup \ell :: \mathcal{H}_\infty \cup r :: \mathcal{H}_\infty \end{aligned}$$

Among all the fixpoints of the above equation, \mathcal{H}_0 is the least fixpoint and describes the inductive type associated with this equation, that is the type of the finite histories and \mathcal{H}_∞ is the greatest fixpoint and describes the coinductive type associated with this equation, that is the type of the infinite and infinite histories. The principle that says that given an equation, the least fixpoint is the inductive type associated with this equation and the greatest fixpoint is the coinductive type associated with this equation is very general and will be used all along this book.

The COQ vernacular, is more verbose, but also more precise in describing the **CoInductive** type *History*, (see Appendix B.3 for a precise definition). The word **coinductive** guarantees that we define actually infinite objects and attach to the objects of type *History* a specific form of reasoning, called *coinduction*. In coinduction, we assume that we “know” an infinite object by observing it through its definition, which is done by a kind of peeling. Since on infinite objects there is no concept of being smaller, one does not reason by saying “I know that the property holds on smaller objects let us prove it on the object”. On the contrary one says “Let us prove a property on an infinite object. For that peel the object, assume that the property holds on the peeled object and prove that it holds on the whole object”. One does not say that the object is smaller, just that the property holds on the peeled object. The above presentation is completely informal, but it has been, formally founded in the theory of COQ, after the pioneer works of Park (1981) and Milner (1989), using the concept of greatest fixpoint in type theory (Coquand, 1993) (see Sangiorgi (2009) for a survey). Bertot and Castéran (2004) present the concepts in Chapter 13 of their book.

Bisimilarity

By just observing them, one cannot prove that two objects which have exactly the same behavior are equal, we can just say that they are observably equivalent. Observable equivalence is a relation weaker than equality⁵, called *bisimilarity*

⁵We are talking here about *Leibniz equality*, not about *extensional equality* see appendix A.1.

and defined on *History* as a *CoInductive* (see Appendix B.3 for a fully formal definition in the *COq* vernacular):

Bisimilarity \sim_h on *histories* is defined **coinductively** as follows:

- $[] \sim_h []$,
- $h \sim_h h'$ implies $\forall a : \text{Agent}, a :: h \sim_h a :: h'$.

This means that two histories are bisimilar if either both are null or for composed histories, if both have the same head and the rests of both histories are bisimilar.⁶ One can prove that two objects that are equal are bisimilar, but not the other way around, because for two objects to be equivalent by observation, does not mean that they have the same identity. To illustrate the difference between bisimilarity and equality of infinite objects let us consider for example two infinite histories α_0 and β_0 that are obtained as solutions of two equations. Let $\alpha_n = c(n) :: \alpha_{n+1}$, where $c(n)$ is (**if** *even*(n) **then** ℓ **else** r), and β_p is $\ell :: r :: \beta_{p+1}$. We know that if we ask for the 5th element of α_0 and β_0 we will get ℓ in both cases, and the $2p^{\text{th}}$ element will be r in both cases, but we have no way to prove that α_0 and β_0 are equal, i.e., have exactly the same structure. Actually the picture in Figure 2.2 shows that they look different and there is no hope to prove by induction, for instance, that they are the same, since they are not well-founded. We see that in the first history, for all p , we have $\alpha_0 = \alpha_{2p+1}$ and, we can see α_0 as fixpoint of the system of equations:

$$x_\alpha = \ell :: x_{\alpha'} \quad x_{\alpha'} = r :: x_\alpha$$

and for the second history, for all p we have also $\beta_0 = \beta_{2p+1}$ and we can see β_0 as the fixpoint of the equation

$$y_\beta = \ell :: r :: y_\beta.$$

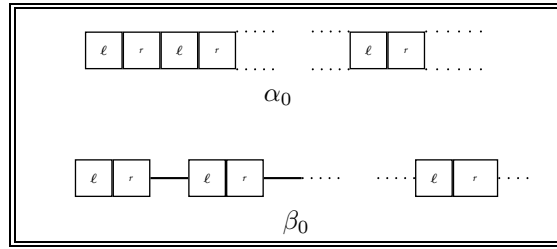


Figure 2.2: The picture of two bisimilar histories

As we said, we do not consider equality among infinite objects, but only bisimilarity. Why? The reason is that with the kind of reasoning we use, we can only prove that two objects are bisimilar, not that they are equal.⁷

⁶Bisimilarity is related with *p-morphisms* and *zigzag relations* in modal logic. See (Sangiorgi, 2009) for a survey.

⁷We can consider a definition of infinite object where objects are equal if they have the same elements, but such a set of objects is obtained by quotient of the set of histories by the

Always

A property P can be true on an infinite history. For instance “*there exists an ℓ in the sequence*”. But we can also say that a property of a history is always true, that is true for all the sub-histories of the history. For instance, “*there is always an ℓ further in the sequence*”, this also means “*there exists infinitely many ℓ 's in the history*”. The operator which transforms a property P in a property *always* P is called a *modality*. The modality *always* is written \square and we write $\square P$ instead of *always* P .

2.2.2 Lazy binary trees

As an example of a coinductive definition consider *lazy binary tree*, i.e., the type of finite and infinite binary trees.

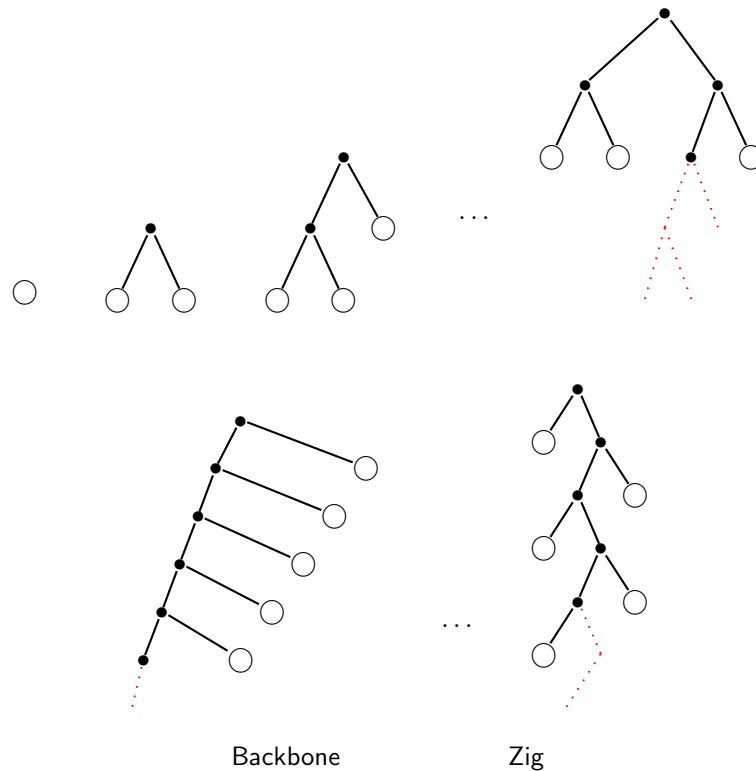


Figure 2.3: Coinductive binary trees

bisimilarity relation. This way, we loose the structure of the infinite objects as we described them. For us it is important to keep the structure of the objects.

A *coinductive binary tree* (or a lazy binary tree or a finite-infinite binary tree) is

- either the empty binary tree \bigcirc ,
- or a binary tree of the form $t \cdot t'$, where t and t' are binary trees.

By the keyword *coinductive* we mean that we define a coinductive set of objects, hence we accept infinite objects. Some coinductive binary trees are given on Fig. 2.3. We define on coinductive binary trees a *predicate* which has also a coinductive definition:

A binary tree is *infinite* if (coinductively)

- either its left subtree is *infinite*
- or its right subtree is *infinite*.

Can we speak about a specific infinite tree? Yes provided we can define it. This can be done as a fixpoint, actually a cofixpoint since we speak about an infinite object. Let us define an infinite binary tree with an infinite path that goes left, then right, then left, then right, then left, forever (Figure 2.3). We call *zig* this infinite tree. Its definition goes with another infinite tree called *zag*.

We define two trees that we call *zig* and *zag*.

zig and *zag* are defined together as *cofixpoints* as follows:

- *zig* has \bigcirc as left subtree and *zag* as right subtree,
- *zag* has *zig* as left subtree and \bigcirc as right subtree.

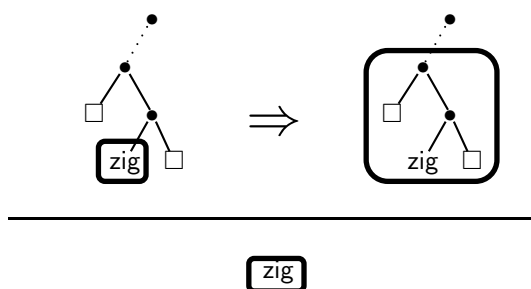
This says that *zig* and *zag* are the greatest solutions⁸ of the two simultaneous equations:

$$\begin{aligned} \text{zig} &= \bigcirc \cdot \text{zag} \\ \text{zag} &= \text{zig} \cdot \bigcirc \end{aligned}$$

It is common sense that *zig* and *zag* are infinite, but to prove that “*zig is infinite*” using the *cofix* tactic⁹, we do as follows: assume “*zig is infinite*”, then *zag* is infinite, from which we get that “*zig is infinite*”. Since we use the assumption on a strict subtree of *zig* (the direct subtree of *zag*, which is itself a direct subtree of *zig*) we can conclude that the *cofix* tactic has been used properly and that the property holds, namely that “*zig is infinite*”. We have proved that “*zig is infinite*” is an invariant along the infinite binary trees *zig* and *zag*. The *cofix* reasoning is pictured on Fig.2.4, where the square box represents the predicate *is infinite*. Above the rule, there is the step of coinduction and

⁸In this case, the least solutions are uninteresting as they are objects nowhere defined. Indeed there is no basic case in the inductive definition.

⁹The *cofix* tactic is a method proposed by the proof assistant COQ which implements coinduction on cofixpoint objects. Roughly speaking, it attempts to prove that a property is an *invariant*, by proving it is preserved along the infinite object. Here “*is infinite*” is such an invariant on *zig*.

Figure 2.4: How cofix works on zig for *is infinite*?

below the rule the conclusion, namely that the whole zig is infinite. We let the reader prove that “*backbone is infinite*”, where *backbone* is the greatest fixpoint of the equation:

$$\text{backbone} = \text{backbone} \cdot \bigcirc$$

Interested readers may have a look at Coupet-Grimal (2003); Coupet-Grimal and Jakubiec (2004); Lescanne (2009); Bertot (2005, 2007) and especially Bertot and Castéran (2004, chap. 13) for other examples of cofix reasoning.

Chapter 3

Games and strategy profiles

We start with a formal and inductive presentation of finite games, which is extended in the next section to a description of infinite games. The chapter ends with a presentation of equilibria in infinite games: Nash equilibria and subgame perfect equilibria.

In classical textbooks, finite and infinite games are presented through their histories. But in the framework of a proof assistant or just to make rigorous proofs, it makes sense to present them structurally. Therefore, games are rather naturally seen as either a leaf to which a *utility function* (a function that assigns a utility, a payoff or a cost to each agent, aka an *outcome*) is attached or a node which is associated to an agent and two subgames. If agents are ALICE and BOB and utilities are natural numbers, a utility function can be the function $\text{ALICE} \mapsto 3, \text{BOB} \mapsto 2$.

3.1 Finite Games

We restrict to infinite games in which each player has two choices at each turn. Such finite extensive game can be seen as built by putting together a player a and two games g_ℓ and g_r , which correspond to either choice made by the player. We write $\langle a, g_\ell, g_r \rangle$ this game. We need also a base case, which is actually what is seen usually as the “end” of a game and which is used here as the basis which every finite game is based upon. Actually it is a degenerated game where players do not play but just receive their payoffs. Assume there are two players ALICE and BOB and p_A is the payoff for ALICE and p_B is the payoff for BOB. This is the utility function $f \equiv \text{ALICE} \mapsto p_A, \text{BOB} \mapsto p_B$. We write $\langle f \rangle$ this kind of game. A finite binary game is a game obtained by applying repeatedly applications of $\langle a, -, - \rangle$ to games of the form $\langle f \rangle$. In other words:

The type *Finite Game* is defined as an **inductive** as follows:

- a *Utility function* makes a *Finite Game*,
- an *Agent* and two *Finite Games* make a *Finite Game*.

Hence one builds a finite game in two ways: either a given utility function f is encapsulated to make the game $\langle f \rangle$, or an agent a and two games g_l and g_r are given to make the game $\langle a, g_l, g_r \rangle$. Notice that in such games, it can be the case that the same agent a has the turn twice in a row, like in the game $\langle a, \langle a, g_1, g_2 \rangle, g_3 \rangle$.

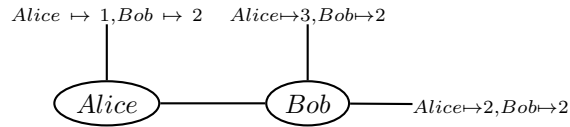
3.2 Infinite Games

We study games that “can” be infinite and “can” have finite or infinite branches, like the 0, 1 game.

The type *Game* is defined as a **coinductive** as follows:

- a *Utility function* makes a *Game*,
- an *Agent* and two *Games* make a *Game*.

A *Game* is either a leaf (a terminal node) or a composed game made of an agent (the agent who has the turn) and two subgames (the formal definition in the COQ vernacular is given in the appendix B.3). Like for finite games, we use the expression $\langle f \rangle$ to denote the leaf game associated with the utility function f and the expression $\langle a, g_l, g_r \rangle$ to denote the game with agent a at the root and two subgames g_l and g_r . For instance, the game we would draw:



is represented by the term:

$$\langle \text{ALICE}, \langle \text{Alice} \mapsto 1, \text{Bob} \mapsto 2 \rangle, \langle \text{BOB} \langle \text{Alice} \mapsto 2, \text{Bob} \mapsto 2 \rangle, \langle \text{Alice} \mapsto 3, \text{Bob} \mapsto 2 \rangle \rangle \rangle$$

Concerning comparisons of utilities we consider a very general setting where a utility is no more than a type (a “set”) with a preference which is a preorder, i. e., a transitive and reflexive relation, and which we write \preceq . A preorder is enough for what we want to prove. By using a very general preorder, it makes extremely easy to go from payoff to cost, we have just to switch the direction of \preceq keeping the same carrier. We assign to the leaves, a utility function which associates a utility to each agent.

Like for histories, to describe an infinite game one uses a fixpoint equation. For instance to describe the 0, 1 game one uses the equation:

$$\text{zero_one} = \langle \text{ALICE}, \langle \text{BOB}, \text{zero_one}, \langle \text{ALICE} \mapsto 0, \text{BOB} \mapsto 1 \rangle \rangle, \langle \text{ALICE} \mapsto 1, \text{BOB} \mapsto 0 \rangle \rangle$$

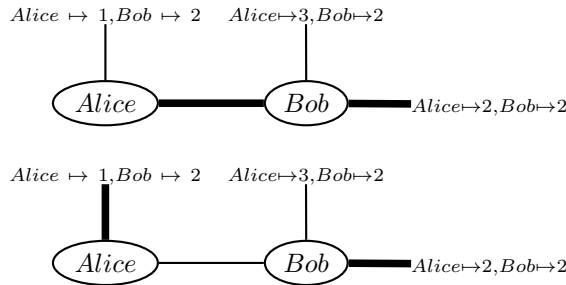
3.3 Infinite Strategy Profiles

The main concept of this book is this of infinite strategy profile which is a coinductive. More specifically, in this book, we focus on infinite binary strategy profiles associated with infinite binary games.

The type of *Strategy Profiles* is defined as a **coinductive** as follows:

- a *Utility function* makes a *Strategy Profile*.
- an *Agent*, a *Choice* and two *Strategy Profiles* make a *Strategy Profile*.

Basically¹ an infinite strategy profile which is not a leaf is a node with four items: an agent, a choice, two infinite strategy profiles. A strategy profile is the same as a game, except that there is a choice. In what follows, since we consider equilibria, we only address strategy profiles. Strategy profiles of the first kind are written $\ll f \gg$ and strategy profiles of the second kind are written $\ll a, c, s_l, s_r \gg$. In other words, if between the “ \ll ” and the “ \gg ” there is one component, this component is a utility function and the result is a leaf strategy profile and if there are four components, this is a node strategy profile. For instance, with the game of page 22 one can associate at least the following strategy profiles:



which correspond to the expressions

$$\ll \text{ALICE}, r, \ll \text{Alice} \mapsto 1, \text{Bob} \mapsto 2 \gg, \ll \text{BOB}, \ell, \ll \text{Alice} \mapsto 2, \text{Bob} \mapsto 2 \gg, \ll \text{Alice} \mapsto 3, \text{Bob} \mapsto 2 \gg \gg$$

and

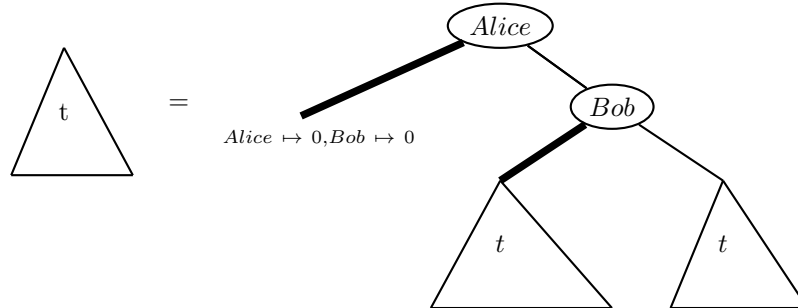
$$\ll \text{ALICE}, \ell, \ll \text{Alice} \mapsto 1, \text{Bob} \mapsto 2 \gg, \ll \text{BOB}, \ell, \ll \text{Alice} \mapsto 3, \text{Bob} \mapsto 2 \gg, \ll \text{Alice} \mapsto 2, \text{Bob} \mapsto 2 \gg, \gg.$$

Let us call s_0 the first strategy profile and s_1 the second one. To describe an infinite strategy profile one uses most of the time a fixpoint equation like:

$$t = \ll \text{Alice}, \ell, \ll \text{Alice} \mapsto 0, \text{Bob} \mapsto 0 \gg, \ll \text{Bob}, \ell, t, t \gg \gg$$

¹The formal definition in the Coq vernacular is given in appendix B.3.

which corresponds to the pictures:



Other examples of infinite strategy profiles are given in Chapter 4. Usually an infinite game is defined as a cofixpoint, i.e., as the solution of an equation, possibly a parametric equation.

Whereas in the finite case we can easily associate with a strategy profile a utility function, i.e., a function which assigns a utility to an agent, as the result of a recursive evaluation, this is no more the case with infinite strategy profiles. One reason is that we are not sure that such a utility function exists for the strategy profile. This makes the function partial. Therefore $s2u$ (an abbreviation for *strategy profile-to-utility*) is a relation between a strategy profile and a utility function, which is also a coinductive; $s2u$ appears in expression of the form $(s2u\ s\ a\ u)$ where s is a strategy profile, a is an agent and u is a utility. It reads “ u is a utility of the agent a in the strategy profile s ”.

$s2u$ is a predicate defined **coinductively** as follows:

- $s2u\ \ll f \gg\ a\ (f(a))$ holds,
- if $s2u\ s_l\ a\ u$ holds then $s2u\ \ll a', \ell, s_l, s_r \gg\ a\ u$ holds,
- if $s2u\ s_r\ a\ u$ holds then $s2u\ \ll a', r, s_l, s_r \gg\ a\ u$ holds.

This means the utility of a for the leaf strategy profile $\ll f \gg$ is $f(a)$, i.e., the value delivered by the function f when applied to a . The utility of a for the strategy profile $\ll a', \ell, s_l, s_r \gg$ is u if the utility of a for the strategy profile s_l is u . For s_0 , the first above strategy profile, one has $s2u\ s_0\ \text{ALICE } 2$, which means that, for the strategy profile s_0 , the utility of ALICE is 2.

3.3.1 The predicate “leads to a leaf”

In order to insure that $s2u$ has a result we define a predicate “leads to a leaf” that says that if one follows the choices shown by the strategy profile one reaches a leaf, i.e., one does not go forever.

The predicate “leads to a leaf” is defined **inductively** as

- the strategy profile $\ll f \gg$ “leads to a leaf”,
- if s_l “leads to a leaf”, then $\ll a, \ell, s_l, s_r \gg$ “leads to a leaf”,

- if s_r “leads to a leaf”, then $\ll a, r, s_l, s_r \gg$ “leads to a leaf”.

This means that a strategy profile which is itself a leaf “leads to a leaf” and if the strategy profile is a node, if the choice is ℓ and if the left strategy subprofile “leads to a leaf” then the whole strategy “leads to a leaf” and similarly if the choice is r . We claim that this gives a good notion of *finite horizon* which seems to be rather a concept on strategy profiles than on games.

If s is a strategy profile that satisfies the predicate “leads to a leaf” then the utility exists and is unique, in other words:

- *Existence.* For all agent a and for all strategy profile s , if s “leads to a leaf” then there exists a utility u which “is a utility of the agent a in the strategy profile s ”.
- *Uniqueness.* For all agent a and for all strategy profile s , if s “leads to a leaf”, if “ u is a utility of the agent a in the strategy s ” and “ v is a utility of the agent a in the strategy s ” then $u = v$.

We say “the” utility in this case since the relation $s2u a$ is functional.

3.3.2 The predicate “always leads to a leaf”

We also consider a predicate “always leads to a leaf” which means that everywhere in the strategy profile, if one follows the choices, one leads to a leaf. This property is defined everywhere on an infinite strategy profile and is therefore coinductive.

The predicate “always leads to a leaf” is defined **coinductively**

- the strategy profile $\ll f \gg$ “always leads to a leaf”,
- for all choice c , if $\ll a, c, s_l, s_r \gg$ “leads to a leaf”, if s_l “always leads to a leaf”, if s_r “always leads to a leaf”, then $\ll a, c, s_l, s_r \gg$ “always leads to a leaf”.

This says that a strategy profile, which is a leaf, “always leads to a leaf” and that a composed strategy profile inherits the predicate from its strategy subprofiles provided itself “leads to a leaf”.

3.3.3 The \Box modality

\Box is a modality, borrowed from temporal logic, i.e., an operator which modifies a predicate. $\Box P$ reads *always P*.

The modality \Box is defined **coinductively** by

- $P \ll f \gg \Rightarrow (\Box P) \ll f \gg$
- $P \ll a, c, s'_l, s'_r \gg \Rightarrow (\Box P)_{s_l} \Rightarrow (\Box P)_{s_f} \Rightarrow (\Box P) \ll a, c, s'_l, s'_r \gg$

One has the proposition:

Proposition 1 $\forall s, (\Box \text{“leads to a leaf”}) s \Leftrightarrow s \text{“always leads to a leaf”}$.

3.3.4 The bisimilarity

We define also bisimilarity between games and between strategy profiles. For strategy profiles, this is defined by:

The bisimilarity \sim_s on *strategy profiles* is defined **coinductively** as follows:

- $\langle\langle f \rangle\rangle \sim_s \langle\langle f \rangle\rangle$,
- if $s_l \sim_s s'_l$ and $s_r \sim_s s'_r$ then $\langle\langle a, c, s_l s_r \rangle\rangle \sim_s \langle\langle a, c, s'_l, s'_r \rangle\rangle$.

This says that two leaves are bisimilar if and only if they have the same utility function and that two strategy profiles are bisimilar if and only if they have the same head agent, the same choice and bisimilar strategy subprofiles.

3.3.5 The game of a strategy profile

We can associate with a strategy profile a game that is the game underlying the strategy profile. In other words, $s2g(s)$ is the game in which all the choices are removed.

The function $s2g$ is defined **coinductively** as follows:

- $s2g \langle\langle f \rangle\rangle = \langle f \rangle$
- $s2g \langle\langle a, c, s_l, s_r \rangle\rangle = \langle a, s2g(s_l), s2g(s_r) \rangle$.

3.4 Subgame perfect and Nash equilibria

Nash equilibria are specific strategy profiles, but to define them one needs the concept of convertibility.

3.4.1 Convertibility

Despite it is not strictly defined in textbooks as such, *convertibility* is an important binary relation on strategy profiles, necessary to speak formally about equilibria. Indeed in order to characterize a strategy profile e as a Nash equilibria, it is assumed that each agent compares the payoff returned by that strategy profile e with the payoff returned by other strategy profiles, which are “converted” from the current one by the agent changing his mind. Since this relation plays a crucial role in formal definition of a Nash equilibrium, it is worth describing, first informally, then a little more formally, knowing that the ultimate formal definition is given in the in COQ vernacular on page 49. Convertibility was introduced for finite games by Vestergaard (2006).

Convertibility informally

Osborne (2004, Chap. 5) presents a Nash equilibrium as “a strategy profile from which no player wishes to deviate, given the other player’s strategies”. We have therefore to say what one means by “a player deviating when the others do not”. In other words, we want to make precise the concept of “deviation”. For that, assume given an agent a and a strategy profile s , a strategy profile in which only finitely many choices made by the given agent a are changed is a “deviation” of a and is said, in this framework, to be convertible to s for a . The binary relation between two strategy profiles, which we call *convertibility*, can be made precise by giving it an inductive definition.² In the previous examples, s_0 is convertible to s_1 for ALICE, since the only change between s_0 and s_1 is ALICE changing her first choice.

Convertibility as an inductively defined mathematical relation

We write $\vdash^a \dashv$ the convertibility for agent a .

The relation $\vdash^a \dashv$ is defined **inductively** as follows:

ConvRef: $\vdash^a \dashv$ is reflexive, i.e., for all s , $s \vdash^a \dashv s$.

ConvAgent : If the node has the same agent as the agent in $\vdash^a \dashv$ then the choice may change, i.e.,

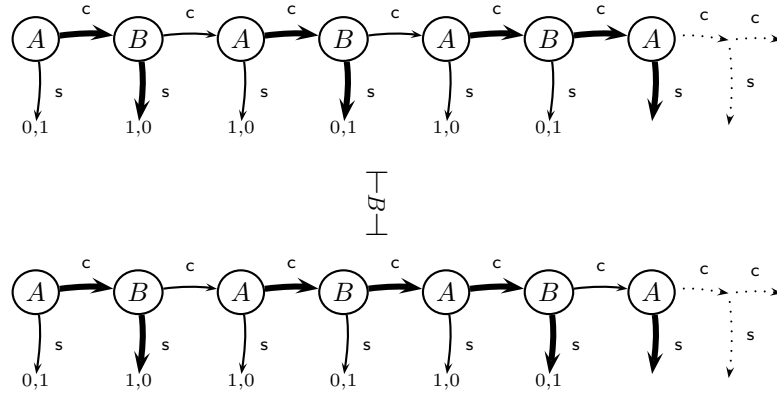
$$\frac{s_1 \vdash^a \dashv s'_1 \quad s_2 \vdash^a \dashv s'_2}{\ll a, c, s_1, s_2 \gg \vdash^a \dashv \ll a, c', s'_1, s'_2 \gg} \text{ConvAgent}$$

ConvChoice: If the node does not have the same agent as in $\vdash^a \dashv$, then the choice has to be the same:

$$\frac{s_1 \vdash^a \dashv s'_1 \quad s_2 \vdash^a \dashv s'_2}{\ll a', c, s_1, s_2 \gg \vdash^a \dashv \ll a', c, s'_1, s'_2 \gg} \text{ConvChoice}$$

Roughly speaking two strategy profiles are convertible for a if their difference only for the choices of a . In the previous example (Section 3.3) we may write $s_0 \vdash^{\text{ALICE}} \dashv s_1$, to say that s_0 is convertible to s_1 for ALICE. In Figure 3.1, page 31, we develop the skeleton of the proof of this convertibility, namely we give the proof tree of $s_0 \vdash^{\text{ALICE}} \dashv s_1$. Since $\vdash^a \dashv$ is defined inductively, this means that the changes are finitely many. We feel that this makes sense since an agent can only conceive finitely many issues. For instance for two strategy profiles associated with the 0, 1 game, we get

²It is also possible to give it a coinductive definition, in which infinitely many choices can be changed, but we feel that this goes beyond the ability of a rational agent who has finite capacities to compute.



3.4.2 Nash equilibria

The notion of Nash equilibrium is translated from the notion in textbooks. Let us recall it. According to Osborne (2004, chap. 5), *A Nash equilibrium is a “pattern[s] of behavior with the property that if every player knows every other player’s behavior she has not reason to change her own behavior”* in other words, “*a Nash equilibrium [is] a strategy profile from which no player wishes to deviate, given the other player’s strategies.*” As we said, the informal concept of deviation is expressed formally by the binary relation “convertibility”. The concept of Nash equilibrium is based on a comparison of utilities. s is a *Nash equilibrium* if the following implication holds:

If for all agent a and for all strategy profile s' which is convertible to s , i.e., $s \vdash^a \neg s'$, if u is the utility of s for a and u' is the utility of s' for a , then $u' \preceq u$.

Roughly speaking this means that a Nash equilibrium is a strategy profile in which no agent has interest to change his choice since doing so he cannot get a better payoff.

3.4.3 Subgame Perfect Equilibria

Let us consider now *subgame perfect equilibria*, which we write *SGPE*. *SGPE* is a property of strategy profiles. It requires the strategy subprofiles to fulfill coinductively the same property, namely to be a *SGPE*, and to insure that the strategy profile with the best utility for the node agent to be chosen. Since both the strategy profile and its strategy subprofiles are potentially infinite, it makes sense to define *SGPE* coinductively.

SGPE is defined **coinductively** as follows:

- $SGPE \ll f \gg$,
- if $\ll a, \ell, s_l, s_r \gg$ “*always leads to a leaf*”, if $SGPE(s_l)$ and $SGPE(s_r)$, if $s_2u s_l a u$ and $s_2u s_r a v$, if $v \preceq u$ then $SGPE \ll a, \ell, s_l, s_r \gg$,

- if $\ll a, r, s_l, s_r \gg$ “always leads to a leaf”, if $SGPE(s_l)$ and $SGPE(s_r)$, if $s_2u s_l a u$ and $s_2u s_r a v$, if $u \preceq v$ then $SGPE \ll a, r, s_l, s_r \gg$,

This means that a strategy profile, which is a leaf, is a subgame perfect equilibrium. Moreover if the strategy profile is a node, if the strategy profile “always leads to a leaf”, if it has agent a and choice ℓ , if both strategy subprofiles are subgame perfect equilibria and if the utility of the agent a for the right strategy subprofile is less than this for the left strategy subprofile then the whole strategy profile is a subgame perfect equilibrium and vice versa. If the choice is r this works similarly.

Notice that since we require that the utility can be computed not only for the strategy profile, but for the strategy subprofiles and for the strategy subprofiles and so on, we require these strategy profiles not only to “lead to a leaf” but to “always lead to a leaf”.

We define orders (one for each agent a) between strategy profiles that lead to a leaf which we write \leq_a .

$s' \leq_a s$ iff : If u (respectively u') is the utility for a in s (resp. s'), then $u' \preceq u$

We say “the” utility since in this case the relation $s_2u a$ is functional.

Proposition 2 \leq_a is an order on strategy profiles which lead to a leaf.

The proof is straightforward.

Proposition 3 A subgame perfect equilibrium is a Nash equilibrium.

Proof: Suppose that s is a strategy profile which is a $SGPE$ and which has to be proved to be is a Nash equilibrium.

Assuming that s' is a strategy profile such that $s \vdash^a \dashv s'$, let us prove by induction on $s \vdash^a \dashv s'$ that $s' \leq_a s$:

- Case $s = s'$, by reflexivity, $s' \leq_a s$.
- Case $s = \ll x, \ell, s_l, s_r \gg$ and $s' = \ll x, \ell, s'_l, s'_r \gg$ with $x \neq a$. $s \vdash^a \dashv s'$ and the definition of $\vdash^a \dashv$ imply $s_l \vdash^a \dashv s'_l$ and $s_r \vdash^a \dashv s'_r$. s_l which is a strategy subprofile of a $SGPE$ is a $SGPE$ as well. Hence by induction hypothesis, $s'_l \leq_a s_l$.

The utility of s (respectively of s') for a is the utility of s_l (respectively of s'_l) for a , then $s' \leq_a s$.

- The case $s = \ll x, r, s_l, s_r \gg$ and $s' = \ll x, r, s'_l, s'_r \gg$ is similar.
- Case $s = \ll a, \ell, s_l, s_r \gg$ and $s' = \ll a, r, s'_l, s'_r \gg$, then $s_l \vdash^a \dashv s'_l$ and $s_r \vdash^a \dashv s'_r$. Since s is a $SGPE$, $s_r \leq_a s_l$.

Moreover, since s_r is a $SGPE$, by induction hypothesis, $s'_r \leq_a s_r$.

Hence, by transitivity of \leq_a , $s'_r \leq_a s_l$. But we know that the utility of s' for a is the one of s'_r and the utility of s for a is the one of s_l , hence $s' \leq_a s$.

- The case $s = \ll a, r, s_l, s_r \gg$ and $s' = \ll a, l, s'_l, s'_r \gg$ is similar.

□

The above proof is a presentation of the formal proof written with the help of the proof assistant COQ. Notice that it is by induction on $\vdash^a \dashv$ which is possible since $\vdash^a \dashv$ is inductively defined. Notice also that s and s' are potentially infinite.

$$\begin{array}{c}
 \frac{}{\ll \text{ALICE} \mapsto 0, \text{BOB} \mapsto 1 \gg \vdash_{\text{ALICE}} \neg \ll \text{ALICE} \mapsto 0, \text{BOB} \mapsto 1 \gg} \text{ConvRef1} \quad \frac{}{\ll \dots \gg \vdash_{\text{ALICE}} \neg \ll \dots \gg} \text{ConvRef1} \\
 \frac{}{\ll \text{BOB}, \ell, \ll \dots \gg \ll \dots \gg \gg \vdash_{\text{ALICE}} \neg \ll \text{BOB}, \ell, \ll \dots \gg \ll \dots \gg \gg} \text{ConvAgent} \quad \frac{}{\ll \text{ALICE} \mapsto 2, \text{BOB} \mapsto 0 \gg \vdash_{\text{ALICE}} \neg \ll \text{ALICE} \mapsto 2, \text{BOB} \mapsto 0 \gg} \text{ConvRef1} \\
 \frac{}{\ll \text{ALICE}, \ell, \ll \text{BOB}, \ell, \ll \dots \gg, \ll \dots \gg \gg, \ll \dots \gg \gg \vdash_{\text{ALICE}} \neg \ll \text{ALICE}, \ell, \ll \text{BOB}, \ell, \ll \dots \gg, \ll \dots \gg \gg, \ll \dots \gg \gg} \text{ConvChoice}
 \end{array}$$

Figure 3.1: An inductive proof of convertibility

3.5 Escalation

A game is susceptible to escalation or not. Obviously the possibility of an escalation in a game requires the game to be infinite.

3.5.1 Escalation informally

Escalation is a property of an infinite game, which says that a game can contain an infinite path along which players always act rationally. In other words, it says that at each turn in the game, there exists a strategy profile which is a subgame perfect equilibrium, in which the player who has the turn continues. Since at each turn, continuing is rational for each player, this means that there is a possibility for players acting rationally to continue forever. That there is an infinite path means that there exists an infinite sequence of games which are direct subgames of their predecessors. For each game of this sequence, there exists a strategy profile which has this game as a skeleton, which is a subgame perfect equilibrium and in which the player who has the turn continues.

3.5.2 Escalation as a formal property

The property of having an escalation can be formalized. A game g has an *escalation* if there exists an escalation sequence $(\mathbf{g}_n)_{n \in \mathbb{N}}$ which is a sequence of subgames of g along the escalation with the following properties: for all n there are two strategy profiles s and s' and an agent a such that

- the game associated with the strategy profile $\langle\langle a, l, s, s' \rangle\rangle$ is bisimilar to the game \mathbf{g}_n and is a *subgame perfect equilibrium* or the game associated with the strategy profile $\langle\langle a, r, s', s \rangle\rangle$ is bisimilar to the game \mathbf{g}_n and is a *subgame perfect equilibrium*
- the game associated with the strategy profile s is bisimilar to the game \mathbf{g}_{n+1} , (this insures that \mathbf{g}_{n+1} is bisimilar to a direct subgame of \mathbf{g}_n , more precisely that \mathbf{g}_n is bisimilar to the game $\langle a, \mathbf{g}_{n+1}, s' \rangle$ or to the game $\langle a, s', \mathbf{g}_{n+1} \rangle$, according to the choice made in the above condition).

This can be made completely formal, by writing it in COQ:

Definition *has_an_escalation_sequence* ($g_seq: nat \rightarrow Game$): $Prop :=$

$$\begin{aligned} & \forall n: nat, \exists s, \exists s', \exists a, \\ & (s2g \langle\langle a, l, s, s' \rangle\rangle \sim_g g_seq\ n \wedge SGPE \langle\langle a, l, s, s' \rangle\rangle \vee \\ & s2g \langle\langle a, r, s', s \rangle\rangle \sim_g g_seq\ n \wedge SGPE \langle\langle a, r, s', s \rangle\rangle) \wedge \\ & s2g\ s \sim_g g_seq\ (n+1). \end{aligned}$$

Definition *has_an_escalation* ($g: Game$) : $Prop :=$

$$\exists g_seq, (has_an_escalation_sequence\ g_seq) \wedge (g_seq\ 0 = g).$$

Chapter 4

Case studies

In this section we study several kinds of games that have some analogies, especially they have a centipede shape, since they have an infinite backbone (on the “left”) and all the right subgames are leaves. In the two last cases, the utilities go to infinity, but in the second (dollar auction game) the utilities go to $(-\infty, -\infty)$ (costs, i.e., the opposites of utilities, go to $(+\infty, +\infty)$), whereas in the third, (infinipede game) the utilities go to $(+\infty, +\infty)$.

4.1 The 0, 1 game

In the 0, 1 game (Fig. 2.1, p. 14) 0 and 1 are payoffs. The 0, 1 game has many subgame perfect equilibria, namely the strategy profiles in which Alice continues always and Bob stops infinitely often and the strategy profiles in which Bob continues always and Alice stops infinitely often.

4.1.1 Two simple subgame perfect equilibria

For what we are interested in, we can consider two strategy profiles, one in each category:

- the strategy profile “ALICE *continues always and BOB stops always*”, which we call **z1AcBs** and,
- the strategy profile “ALICE *stops always and BOB continues always*”, which we call **z1AsBc**.

The reasoning to show that **z1AcBs** is a subgame perfect equilibrium works as follows. In this strategy profile ALICE gets 1 and BOB gets 0. Assume the strategy subprofile of **z1AcBs** after the second turn for ALICE¹ is a subgame perfect equilibrium, for which ALICE gets 1 and BOB gets 0. The strategy subprofile that starts at BOB’s turn and which we call **sg_z1AcBs** is a subgame

¹which is nothing but **z1AcBs**!

perfect equilibrium for which ALICE gets 1 and BOB gets 0, since its two strategy subprofiles are subgame perfect equilibria for which ALICE gets 1 and BOB gets 0. `z1AcBs` is a subgame perfect equilibrium since its two strategy subprofiles are two subgame perfect equilibria, one is `sg_z1AcBs` for which ALICE gets 1 and the other is a leaf for which ALICE gets 0.

The same reasoning applies to `z1AsBc` to prove that it is a subgame perfect equilibrium.

4.1.2 Cutting the game and extrapolating

If one cuts the 0,1 game at a finite position, to obtain a finite game, one can cut either after ALICE like on the left below or one can cut after BOB like on the right below:



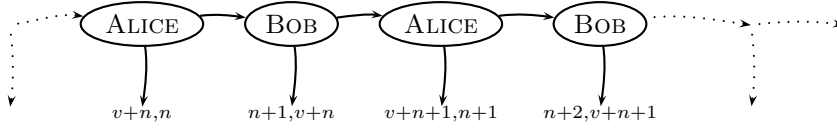
When one cuts after ALICE, the backward induction equilibrium is when ALICE continues always and BOB does whatever he wants and when one cuts after BOB, the backward induction equilibrium is when ALICE does whatever she wants and BOB continues always. One sees that those equilibria cannot be extrapolated to the infinity since they are inconsistent except in the only case when ALICE and BOB continue forever. But this strategy profile cannot be a subgame perfect equilibrium, since it does not fulfill the predicate “*always leads to a leaf*”. Hence cutting the game at a finite position gives no clue on what one obtains at the limit on the infinite game.

4.1.3 The 0,1 game has a rational escalation

One sees that `zero_one` has a rational escalation. Indeed at each step, the agents can always make the choice to continue which is rational since this corresponds to the first choice of a subgame perfect equilibrium. If both agents make the choice to continue, this is the escalation. ALICE continues since she feels that despite BOB did not stop, he will eventually stop and BOB continues because he feels that ALICE will eventually stop.

4.2 The dollar auction game and its Nash equilibria

The dollar auction has been presented by Shubik (1971) as the paradigm of escalation, insisting on its paradoxical aspect. It is a sequential game presented as an auction in which two agents compete to acquire an object of value v ($v > 0$) (see Gintis (2000, Ex. 3.13)). Suppose that both agents bid \$1 at each turn. If one of them gives up, the other receives the object and both pay the

Figure 4.1: The *dollar auction* game

amount of their bid.² For instance, if agent ALICE stops immediately, she pays nothing and agent BOB, who acquires the object, has a payoff v . In the general turn of the auction, if ALICE abandons, she loses the auction and has a payoff $-n$ and BOB who has already bid $-n$ has a payoff $v - n$. At the next turn after ALICE decides to continue, bids \$1 for this and acquires the object due to BOB stopping, ALICE has a payoff $v - (n + 1)$ and BOB has a payoff $-n$. In our formalization we have considered the *dollar auction* up to infinity. Since we are interested only by the “asymptotic” behavior, we can consider the auction after the value of the object has been passed and the payoffs are negative. The dollar auction game can be summarized by Figure 4.1. Notice that we assume that ALICE starts.

4.2.1 Equilibria in the dollar auction

We have recognized three classes of infinite strategy profiles, indexed by n :

1. The strategy profile *always give up*, in which both ALICE and BOB stop at each turn, in short dolAsBs_n .
2. The strategy profile *ALICE stops always and BOB continues always*, in short dolAsBc_n .
3. The strategy profile *ALICE continues always and BOB stops always*, in short dolAcBs_n .

The three kinds of strategy profiles are presented in Figure 4.2.

In the figures like in the COQ implementation, we use *costs*³ instead of payoffs or utilities, since it is simpler in the COQ formalization to reason on natural numbers.

We have shown⁴ that the second and third kinds of strategy profiles, in which one of the agents always stops and the other continues, are subgame perfect equilibria. For instance, consider the strategy profile dolAsBc_n . Assume $\text{SGPE}(\text{dolAsBc}_{n+1})$. It works as follows: if dolAsBc_{n+1} is a subgame perfect equilibrium corresponding to the payoff $-(v + n + 1), -(n + 1)$, then

$$\ll \text{BOB}, \ell, \text{dolAsBc}_{n+1}, \ll \text{ALICE} \mapsto -(n + 1), \text{BOB} \mapsto -(v + n) \gg \gg$$

²In a variant, each bidder, when he bids, puts a dollar bill in a hat or in a piggy bank and there is no return at the end of the auction. The last bidder gets the object.

³Recall that cost is the opposite of utility.

⁴The proofs are typical uses of the Coq *cofix* tactic.

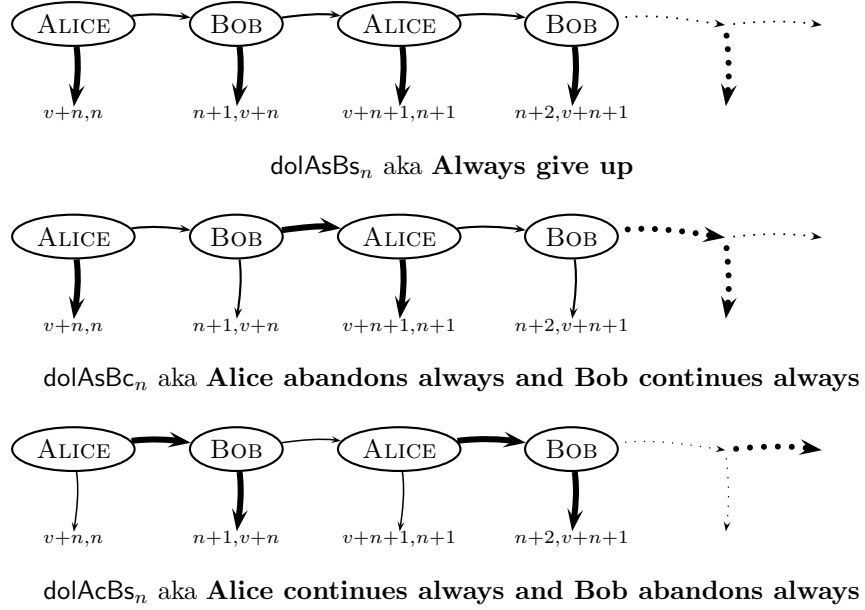


Figure 4.2: Three strategy profiles

is again a subgame perfect equilibrium (since $-(n+1) \geq -(v+n)$) and therefore dolAsBc_n which is

$$\ll \text{ALICE}, r, \ll \text{BOB}, \ell, \text{dolAsBc}_{n+1}, \ll \text{ALICE} \mapsto -(n+1), \text{BOB} \mapsto -(v+n) \gg, \\ \ll \text{ALICE} \mapsto -(v+n), \text{BOB} \mapsto -n \gg \gg$$

is a subgame perfect equilibrium, since again $-(v+n) \geq -(v+n+1)$.⁵ We can conclude that for all n , dolAsBc_n is a *subgame perfect equilibrium*. In other words, we have assumed that $\text{SGPE}(\text{dolAsBc}_n)$ is an *invariant* all along the game and that this invariant is preserved as we proceed backward, through time, into the game.

With the condition $v > 1$, we can prove that dolAsBs_0 is not a Nash equilibrium, then as a consequence not a subgame perfect equilibrium. Therefore, the strategy profile that consists in stopping from the beginning and forever is not a Nash equilibrium, this contradicts what is said in the literature (Shubik, 1971; O’Neill, 1986; Leininger, 1989; Gintis, 2000), with a finite game to approximate an infinite game (the escalation).

4.2.2 Escalation is rational in the dollar auction

Agents are rational when they choose at each step, what they feel to be their best option. Many authors agree⁶ that rationality is choosing a subgame perfect

⁵Since the *cofix* tactic has been used on a strict strategy subprofile, the reasoning is correct.

⁶See however (Halpern, 2001; Stalnaker, 1998).

equilibrium. Aumann (1995) is one of the strongest advocate of this position. His principle in the dollar auction says that a rational agent will choose at each step one of the strategy profiles which is a subgame perfect equilibrium, namely dolAsBc_n or dolAcBs_n . Suppose ALICE is in the middle of the auction, she has two options that are rational: one option is to take BOB's threat seriously and to stop right away, since she assumes that BOB will continue always (strategy profile dolAsBc_n). But in her second option, she admits that from now on BOB will stop always (strategy profile dolAcBs_n) and she will always continue: this is a subgame perfect equilibrium, hence she is rational. If BOB acts similarly this is the escalation. So at each step an agent can stop and be rational, as well as at each step an agent can continue and be rational; both options make sense, and the escalation as well.

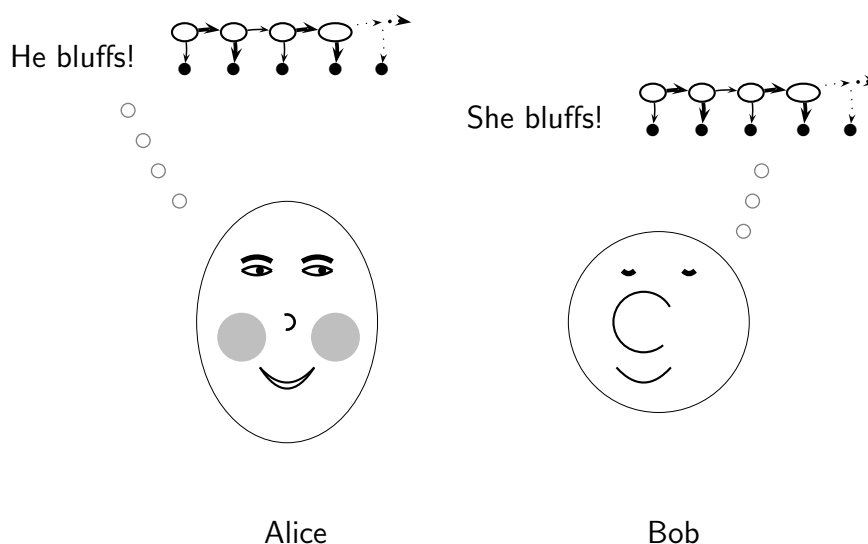


Figure 4.3: What Alice and Bob might think

We claim that human agents reason coinductively unknowingly. Therefore, for them, continuing always is one of their rational options at least if one considers strictly the rules of the dollar auction game with no limit on the bankroll. If at all steps, both agents continue always, this is the escalation. Many experiences (see (Colman, 1999) for a survey) have shown that human are inclined to escalate or at least to go very far in the auction when playing the dollar auction game. We propose the following explanation: the finiteness of the game was not explicit for the participants and for them the game was naturally infinite. Therefore they adopted a kind of reasoning similar to the one we develop here, probably in an intuitive form and they conclude it was equally rational to continue or to leave according to their comprehension of the threat of their

opponent. Actually our theoretical work reconciles experiences with logic,⁷ and human reasoning with rationality.

If agents would have a global view, they would notice that escalation is scarring and ruining and should be avoided. Since it is rational it looks ineluctable, but there are a few ways to avoid it. First, the game can be stopped by the action of an external observer, a kind of master of ceremony who declares the game over, something similar to the finite payroll. Second, since it is a consequence of the absence of communication between agents, it can be avoided, if the agents communicate, like Kennedy and Khrushchev did with the Moscow-Washington hotline. Third, escalation can be stopped by introducing another challenge, like building Europe by Adenauer and de Gaulle.

4.3 The infinipede

Often studied, the extensive game called *centipede*⁸ has been introduced by Rosenthal (1981) (see also Binmore (1987); Colman (1998); Osborne and Rubinstein (1994)). In Rosenthal (1981, p. 96) the game is pictured as shown in Figure 4.4.

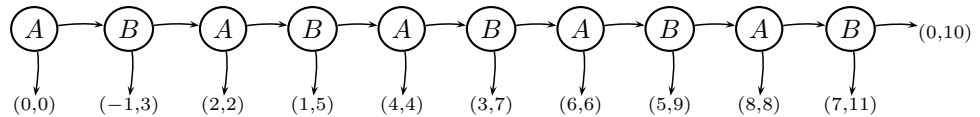


Figure 4.4: The genuine Rosenthal game

This finite game has one Nash equilibrium obtained by backward induction, namely by agent *A* stopping immediately. This game has been extended to hundred, thousand nodes, but all those extensions are finite and all authors conclude that there is a unique Nash equilibrium in which the agents give up immediately.

Since we had noticed, in the case of 0,1 game and the dollar auction, a discrepancy between Nash equilibria when going from finite games to infinite games, it is challenging to see whether the same phenomenon occurs when going from the *centipede* (a finite game) to the *infinipede* (an infinite game).

The *infinipede* is an infinite game in extensive form in which agent *A* has the choice to continue or to end the game with both agents receiving a payoff $2n$ and agent *B* has the choice to continue or to end the game with agent *A* receiving $2n - 1$ and agent *B* receiving $2n + 3$. The generalization is an infinite game which can be pictured as follows (we use subtraction over naturals in which $2 * 0 - 1 = 0$):

In infinipedes, we have identified only one subgame perfect equilibrium, namely this where both agents abandon at each turn (Figure 4.6). We call

⁷A logic which includes coinduction.

⁸A centipede has hundred legs, whereas a millipede has thousand. All belong to the group of myriapods which means “ten thousand legs”.

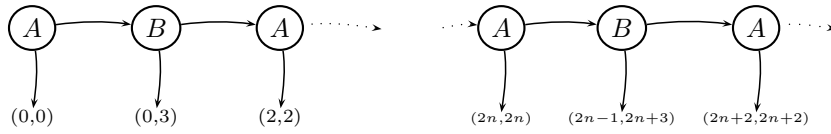


Figure 4.5: The infinipede

it `cent_agu n`. This shows that even in the infinite generalization, agents are

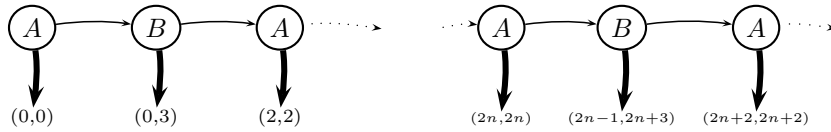


Figure 4.6: Subgame perfect equilibrium for the infinipede

rational if they do not start the game and abandon from the beginning. We have actually shown this property for each value of n , namely for all n , `cent_agu n` is a subgame perfect equilibrium. In COQ we have proved the theorem:

Theorem *SGPE_cent_AGU*: $\forall (n:\text{nat}), \text{SGPE le } (\text{cent_agu } n)$.

Hence the paradox remains: the agents do not get the somewhat better payoff, they would get if they would be more flexible with respect to rationality. However, there remains a problem for the agents in the infinipede game: when they start an infinite game, they do not know when to stop. After all, ending immediately is perhaps a good choice to solve this dilemma.

Chapter 5

Related works

To our knowledge, the only application of coinduction to extensive game theory has been made by Capretta (2007) who uses coinduction to define only common knowledge not equilibria in infinite games. Another strongly connected work is this of Coupet-Grimal (2003) on temporal logic. Other applications are on representation of real numbers by infinite sequences (Bertot, 2007; Julien, 2008) and implementation of streams (infinite lists) in electronic circuits (Coupet-Grimal and Jakubiec, 2004). An ancestor of our description of infinite games and infinite strategy profiles is the constructive description of finite games, finite strategy profiles, and equilibria by Vestergaard (2006). Lescanne (2009) introduces the framework of infinite games. Infinite games are introduced in Osborne and Rubinstein (1994) and Osborne (2004) using histories, but this is not algorithmic and therefore not amenable to formal proofs and coinduction. Exercice 175.1 of Vestergaard (2006) is the dollar auction and no infinite subgame perfect equilibrium is considered.

Many authors have studied infinite games (see for instance Martin (1998); Mazala (2001)), but except the name “game” (an overloaded one), those games have nothing to see with infinite extensive games as presented in this book. The infiniteness of Blackwell games for instance is derived from a topology, by adding real numbers and probability. Sangiorgi (2009) mentioned the connection between Ehrenfeucht-Fraïssé games (Ebbinghaus and Flum, 1995) and coinduction, but the connection with extensive games is extremely remote.

This work started after this of Vestergaard (2006) on finite games and finite strategy profiles. We first developed proofs on finite strategy profiles, but unlike Vestergaard who based his formalization on fixpoint definitions of predicates, we used only inductive definitions of predicates. Like Vestergaard, we were able to prove the main lemma of finite extensive games, namely that *backward induction strategy profiles are Nash equilibria*; the script is available at http://perso.ens-lyon.fr/pierre.lescanne/COQ/INFGAMES/SCRIPTS/finite_games.v.

Overall, this “induction based” presentation allowed us to switch more easily to coinduction on infinite games. Beside this, a development in Coq of finite games with an arbitrary number of choices at any node has been made by

Le Roux (2008, p. 83 and following) and an exploration of common knowledge, induction and Aumann’s theorem on rationality has been proposed by Vestergaard et al. (2006). In Lescanne (2007), there is a presentation of a somewhat connected development in COQ, namely this of the *logic of common knowledge*.

Since we are talking about some computational aspects of games, people may make some analogies with other works, let us state what extensive games are not.

- Extensive games are not *semantic games* as presented in Abramsky and Jagadeesan (1994); Lorenz and Lorenzen (1978); Girard (2001); van Benthem (2006).
- Extensive games are not *logical games* used in proving properties of automata and protocols Merz (2000); Affeldt et al. (2007).
- This work has only loose connection with *algorithmic game theory* Nisan et al. (2007); Daskalakis et al. (2009), which is more interested by the complexity of the algorithms, especially those which compute equilibria, than by their correction, and does not deal with infinite games.
- Extensive games are not Ehrenfeucht-Fraïssé games, but as reported by Sangiorgi (2009) they are related to coinduction through bisimulation.

The book of Dowek (2007) gives a philosophical perspective of using a proof assistant based on type theory in mathematics.

Chapter 6

Postface

Claiming resources are finite leads to reject escalation as non rational.

Taking resources as infinite makes escalation rational.

Here we develop ideas which were not presented in the rest of the book.

6.1 Finite resources vs infinite resources

The preceding discussion shows that the relation between rationality and escalation is connected with the perception of the finiteness of the world and/or the quantity of available resources. Actually people can be split into two categories:

- People who view the resources as finite and take escalation as irrational.
- People who view the resources as infinite and take escalation as rational.

In the first category, we can put the environmentalists, the Club of Rome, Martin Shubik, Al Gore¹ and in the second category the speculators, the gamblers, the risk takers, the Concorde project managers, Bernard Madoff, Kim-Jong-Il. We claim that the pros and cons of infinite resources are the same as the pros and cons of escalation. Perhaps, the reader like the author may think that opting for the first category is wise, but the second option has many fans, since escalation is everywhere in the current world.

Let us consider examples of escalation.

In economy The most amazing example of escalation is *speculative or economic bubble*, a well-known and old phenomenon.

In evolution theory The *red queen hypotheses* and the survival of species. Assume two species are living together and compete for resource. The

¹This division exists among specialiste of set theory, i.e., among those who accept the axiom of well-foundedness (von Neumann, 1928) and those who reject it (Aczel, 1988).

only way for both species to survive is to increase their fitness. This lead to a kind of arm race or a kind of escalation. In other words, for species to survive (in an infinite world), it is necessary to escalate, which is sensible in a presentation of the species competition by game theory.

6.2 Conclusion

Thanks to coinduction, we have reconciled human reasoning with rational reasoning in infinite extensive games. In other words, we claim that human agents reason actually by coinduction when faced to infinite games and are rational. Moreover we have shown once more the threshold between finiteness and infiniteness and that reasoning on infinite objects is not the limit when the size goes to infinity of reasoning on finite objects.

Appendix A

Two subtle points of coinduction

A.1 Equalities

Leibniz equality says that $x = y$ if and only if, for every predicate P , $P(x)$ implies $P(y)$. *Extensional equality* says that $f = g$ if and only if, for all x , $f(x) = g(x)$. In general, knowing a (recursive) definition of f and a (recursive) definition of g is not enough to decide whether $f = g$ or $f \neq g$. For instance, no one knows how to prove that the two functions:

$$\begin{aligned}f(1) &= 1 \\f(2x) &= f(x) \\f(2x + 1) &= f(3x + 2).\end{aligned}$$

and

$$g(x) = 1$$

are equal, despite it is more likely that they are. More generally, there is no algorithm (no rigorous reasoning) which decides whether a given function h is equal to the above function g or not. Thus *extensional equality* is not decidable. Saying that two sequences that have equal elements are equal requires *extensional equality* and it makes sense to reject such an equality when reasoning finitely about infinite objects, like human agents would do.

A.2 Why in infinite runs, agents do not have a utility?

In an infinite play, a play that runs forever, i.e., that does not lead to a leaf, no agent has a utility. People might say that this an anomaly, but this is perfectly sensible. In arbitrary long plays, which lead to a leaf, all agents have a utility.

Only in plays that diverge, it is the case that agents have no utility. This fits well with Binmore (1988) statements “*The use of computing machines (automata) to model players in an evolutive context is presumably uncontroversial ... machines are also appropriate for modeling players in an eductive context*”. Here we are concerned by the eductive context where “*equilibrium is achieved through careful reasoning by the agents before and during the play of the game*” (Binmore, 1988, loc. cit). By automaton, we mean any model of computation¹, since all the models of computation are equivalent by Church thesis. If an agent is modeled by an automaton, this means also that the function that computes the utility for this agent is also modeled by an automaton. It seems then sensible that one cannot compute the utility or the cost of an agent for an infinite play, since computing is a finite process working on finite data (or at least data that are finitely described). Since the agent cannot compute the utility of an infinite play, no sensible value can be attributed to him. If one wants absolutely to assign a value to an infinite play, one must abandon the automaton framework. Moreover this value should be the limit of a sequence of values, which does not exist in most of the cases.² For instance, in the case of the dollar auction (Section 4.2), the costs of ALICE associated with the infinite play are the sequence $\dots, v + n, n + 1, v + n + 1, n + 2, \dots$. In the case of the 0, 1 game the payoff of ALICE is 0, 1, 0, 1, \dots . Therefore considering that in infinite plays, agents have no utility or costs is perfectly consistent with a modeling of agents by automata. By the way, does an agent care about a payoff he (she) receives in infinitely many years? Will he (she) adapt his (her) strategy on this?

¹Our model of computation is this of the calculus of inductive construction, a kind of λ -calculus behind COQ (Turing, 1937).

²If utilities are natural numbers, it exists only if the sequence is stationary, which is not the case in escalation.

Appendix B

About the Coq development

B.1 The notation of functions in the Coq vernacular

In traditional mathematics, the result of applying a function f to the value x is written $f(x)$ and the result of applying f to x and y is written $f(x, y)$, this can be considered as the result of applying f to x then to y and written $f(x)(y)$. In the COQ vernacular, as in type theory, one writes $f x$ instead of $f(x)$ and $f x y$ instead $f(x)(y)$ or instead of $f(x, y)$ and $f x y z$ instead $f(x)(y)(z)$ or $f(x, y, z)$, because this saves parentheses and commas and because the concept of functions is the core of the formalization. But after all, this is just a matter of style and COQ accepts syntactic shorthands to avoid these notations when others are desirable.

B.2 Coinduction in Coq

As we have said the proof assistant COQ (The Coq development team (2007)) played a central role in this research.

Why should we formalize a concept in a proof assistant?

To answer this question we like to cite Donald Knuth (Shustek, 2008):

People have said you don't understand something until you've taught it in a class. The truth is you don't really understand something until you've taught it to a computer, until you've been able to program it.

We claim that we can appropriately replace the last sentence by “*until you’ve taught it to a proof assistant, until you’ve code it into COQ,¹ Isabelle,² or PVS³*” as it seems even more demanding to “teach” a proof assistant like COQ than to write a program on the same topics. Actually without COQ, which has coinduction features, we would not have been able to capture the concepts of Nash equilibrium and Subgame Perfect equilibria presented in this book. This is indeed the result of formal deduction, intuition and try and error in COQ since proving properties of infinite games and infinite strategy profiles is extremely subtle. Moreover by relying on a proof assistant, we can free this article from formal developments and tedious and detailed proofs, knowing anyway that they are correct in any detail and that the reader will refer to the COQ script in case of doubt. Therefore, we can focus on informal explanation. However, COQ proposes a readable, rigorous, and computer checked syntax, the *vernacular*, for definitions, lemmas and theorems and when we provide definitions in this book, they are associated with expressions stated in the vernacular provide in the appendix. The vernacular should be seen as a XXIst century version of Leibniz’ *characterica universalis* or Frege’s *Begriffsschrift* (Frege, 1967).

Decomposing an object

The principle of coinduction is based on the greatest fixpoint of the definition, that is a *coinduction defines a greatest fixpoint* (see (Bertot and Castéran, 2004)). There are two challenges when one works with such a principle: the difficulty of decomposing infinite objects and the invocation of coinduction. They are both presented in detail by Bertot and Castéran (2004), but let us describe them in a few words. For the first problem, suppose one has a strategy profile s , which is not a leaf; one knows that s is of the form $\ll a, c, s_l, s_r \gg$ for some agent a , some choice c and some strategy profiles s_l and s_r . To obtain such a presentation, one uses a mechanism which consists in defining a function identity on strategy profile which is a “clone” of $\text{fun } s \Rightarrow s \text{ end}$:

Definition *Strategy_identity* ($s:\text{Strategy}$): *Strategy* :=
match s *with*
 — $\ll f \gg \Rightarrow \ll f \gg$
 — $\ll a, c, s_l, s_r \gg \Rightarrow \ll a, c, s_l, s_r \gg$
end.

In other words, the *strategy identity* function is, computationally speaking, the function which associates $\ll f \gg$ with $\ll f \gg$ and $\ll a, c, s_l, s_r \gg$ with $\ll a, c, s_l, s_r \gg$ and not the function which associates s with s . We can prove the *strategy decomposition* lemma:

Lemma *Strategy_decomposition*: $\forall s: \text{Strategy},$
Strategy_identity $s = s$.

¹The Coq development team (2007).

²Nipkow et al. (2002).

³Owre et al. (1992).

Thus when one wants to decompose a strategy s , one replaces s by *Strategy_identity* s and one simplifies the expression, and one gets $\ll a, c, s_l, s_r \gg$ for some a, c, s_l and s_r .

Invoking coinduction

The *principle of coinduction* is based on a *tactic*⁴ called *cofix*. It consists in assuming the proposition one wants to proof, provided one applies it only on strict sub-objects. In the current implementation of COQ, the user has to ensure that he invokes it on “strict” sub-objects. This is not always completely trivial and requires a good methodology. However the *proof checker* (a piece of software which accepts only correct proofs) verifies that this constraint is fulfilled at the time of checking the proof.

B.3 Excerpts of the Coq development

The full development is in the url

<http://perso.ens-lyon.fr/pierre.lescanne/COQ/ER/SCRIPTS/>

with a description in

<http://perso.ens-lyon.fr/pierre.lescanne/COQ/ER/HTML/>.

B.3.1 Infinite binary trees

CoInductive *LBintree* : Set :=

— *LbtNil*: *LBintree*

— *LbtNode*: *LBintree* → *LBintree* → *LBintree*.

CoInductive *InfiniteLBT*: *LBintree* → Prop :=

— *IBTLeft* : ∀ *bl br*, *InfiniteLBT bl* → *InfiniteLBT (LbtNode bl br)*

— *IBTRight* : ∀ *bl br*, *InfiniteLBT br* → *InfiniteLBT (LbtNode bl br)*.

CoFixpoint *Zig*: *LBintree* := *LbtNode Zag LbtNil*

with *Zag*: *LBintree* := *LbtNode LbtNil Zig*.

B.3.2 Infinite games

CoInductive *Game* : Set :=

— *gLeaf*: *Utility_fun* → *Game*

— *gNode* : *Agent* → *Game* → *Game* → *Game*.

B.3.3 Infinite strategy profiles

CoInductive *StratProf* : Set :=

— *sLeaf* : *Utility_fun* → *StratProf*

⁴A *tactic* is a tool in COQ used to build proofs without using the most elementary constructions.

— $sNode : Agent \rightarrow Choice \rightarrow StratProf \rightarrow StratProf \rightarrow StratProf$.

Inductive $s2u : StratProf \rightarrow Agent \rightarrow Utility \rightarrow Prop :=$

- $s2uLeaf : \forall a f, s2u (\ll f \gg) a (f a)$
- $s2uLeft : \forall (a a' : Agent) (u : Utility) (sl sr : StratProf),$
 $s2u sl a u \rightarrow s2u (\ll a', l, sl, sr \gg) a u$
- $s2uRight : \forall (a a' : Agent) (u : Utility) (sl sr : StratProf),$
 $s2u sr a u \rightarrow s2u (\ll a', r, sl, sr \gg) a u$.

Lemma $Existence_s2u : \forall (a : Agent) (s : StratProf),$

$LeadsToLeaf s \rightarrow \exists u : Utility, s2u s a u$.

Lemma $Uniqueness_s2u : \forall (a : Agent) (u v : Utility) (s : StratProf),$

$LeadsToLeaf s \rightarrow s2u s a u \rightarrow s2u s a v \rightarrow u = v$.

Inductive $LeadsToLeaf : StratProf \rightarrow Prop :=$

- $LtLLeaf : \forall f, LeadsToLeaf (\ll f \gg)$
- $LtLLeft : \forall (a : Agent) (sl : StratProf) (sr : StratProf),$
 $LeadsToLeaf sl \rightarrow LeadsToLeaf (\ll a, l, sl, sr \gg)$
- $LtLRight : \forall (a : Agent) (sl : StratProf) (sr : StratProf),$
 $LeadsToLeaf sr \rightarrow LeadsToLeaf (\ll a, r, sl, sr \gg)$.

CoInductive $AlwLeadsToLeaf : StratProf \rightarrow Prop :=$

- $ALtLeaf : \forall (f : Utility_fun), AlwLeadsToLeaf (\ll f \gg)$
- $ALtL : \forall (a : Agent) (c : Choice) (sl sr : StratProf),$
 $LeadsToLeaf (\ll a, c, sl, sr \gg) \rightarrow AlwLeadsToLeaf sl \rightarrow AlwLeadsToLeaf sr \rightarrow$
 $AlwLeadsToLeaf (\ll a, c, sl, sr \gg)$.

B.3.4 Convertibility

Inductive $IndAgentConv : Agent \rightarrow StratProf \rightarrow StratProf \rightarrow Prop :=$

- $ConvRef : \forall (a : Agent) (s : StratProf), IndAgentConv a s s$
- $ConvAgent : \forall (a : Agent) (c c' : Choice) (sl sl' sr sr' : StratProf),$
 $(IndAgentConv a sl sl') \rightarrow (IndAgentConv a sr sr') \rightarrow$
 $IndAgentConv a (\ll a, c, sl, sr \gg) (\ll a, c', sl', sr' \gg)$
- $ConvChoice : \forall (a a' : Agent) (c : Choice) (sl sl' sr sr' : StratProf),$
 $IndAgentConv a sl sl' \rightarrow (IndAgentConv a sr sr') \rightarrow$
 $IndAgentConv a (\ll a', c, sl, sr \gg) (\ll a', c, sl', sr' \gg)$.

Notation " $sl \vdash^a \dashv sr$ " := $(IndAgentConv a sl sr)$.

B.3.5 SGPE

CoInductive $SGPE : StratProf \rightarrow Prop :=$

- $SGPE_leaf : \forall f : Utility_fun, SGPE (\ll f \gg)$
- $SGPE_left : \forall (a : Agent) (u v : Utility) (sl sr : StratProf),$
 $AlwLeadsToLeaf (\ll a, l, sl, sr \gg) \rightarrow$
 $SGPE sl \rightarrow SGPE sr \rightarrow$
 $s2u sl a u \rightarrow s2u sr a v \rightarrow (v \preceq u) \rightarrow$

$SGPE \langle\langle a, l, sl, sr \rangle\rangle$
 — $SGPE_right: \forall (a:Agent) (u v:Utility) (sl sr: StratProf),$
 $AlwLeadsToLeaf \langle\langle a, r, sl, sr \rangle\rangle \rightarrow$
 $SGPE \ sl \rightarrow SGPE \ sr \rightarrow$
 $s2u \ sl \ a \ u \rightarrow s2u \ sr \ a \ v \rightarrow (u \preceq v) \rightarrow$
 $SGPE \langle\langle a, r, sl, sr \rangle\rangle.$

B.3.6 Nash equilibrium

Definition $NashEq (s: StratProf): Prop :=$
 $\forall a \ s' \ u \ u', s \vdash^a \neg s \rightarrow (s2u \ s' \ a \ u') \rightarrow (s2u \ s \ a \ u) \rightarrow (u' \preceq u).$

B.3.7 Dollar Auction

Notation " $[x, y]$ " :=
 $(sLeaf (fun a:Alice_Bob \Rightarrow match a with Alice \Rightarrow x \mid Bob \Rightarrow y end))$
 (at level 80).

B.3.8 Alice stops always and Bob continues always

Definition $add_Alice_Bob_dol (cA cB:Choice) (n:nat) (s:Strat) :=$
 $\langle\langle Alice, cA, \langle\langle Bob, cB, s, [n+1, v+n] \rangle\rangle, [v+n, n] \rangle\rangle.$

CoFixpoint $dolAcBs (n:nat): Strat := add_Alice_Bob_dol \ l \ r \ n \ (dolAcBs \ (n+1)).$

Theorem $SGPE_dol_Ac_Bs: \forall (n:nat), SGPE \ ge \ (dolAcBs \ n).$

B.3.9 Alice continues always and Bob stops always

CoFixpoint $dolAsBc (n:nat): Strat := add_Alice_Bob_dol \ r \ l \ n \ (dolAsBc \ (n+1)).$

Theorem $SGPE_dol_As_Bc: \forall (n:nat), SGPE \ ge \ (dolAsBc \ n).$

B.3.10 Always give up

CoFixpoint $dolAsBs (n:nat): Strat := add_Alice_Bob_dol \ r \ r \ n \ (dolAsBs \ (n+1)).$

Theorem $NotSGPE_dolAsBs: (v>1) \rightarrow \sim(NashEq \ ge \ (dolAsBs \ 0)).$

B.3.11 Infinipede

Definition $add_Alice_Bob_cent (cA cB:Choice) (n:nat) (s:Strat) :=$
 $\langle\langle Alice, cA, \langle\langle Bob, cB, s, [2 \times n - 1, 2 \times n + 3] \rangle\rangle, [2 \times n, 2 \times n] \rangle\rangle.$

CoFixpoint $cent_agu (n:nat): (Strat) :=$
 $add_Alice_Bob_cent \ r \ r \ n \ (cent_agu \ (S \ n)).$

Lemma $AlwLeadsToLeaf_cent_agu: \forall (n:nat), AlwLeadsToLeaf \ (cent_agu \ n).$

Lemma $LeadsToLeaf_cent_agu: \forall (n:nat), LeadsToLeaf \ (cent_agu \ n).$

Theorem *SGPE_cent_AGU*: $\forall (n:\text{nat}), \text{SGPE le } (\text{cent_agu } n)$.

Lemma *NashEq_cent_agu*: $\forall (n:\text{nat}), \text{NashEq le } (\text{cent_agu } n)$.

B.3.12 Escalation

Definition *has_an_escalation_sequence* ($g_seq:\text{nat} \rightarrow \text{Game}$): $\text{Prop} := \forall n:\text{nat},$

$\exists s, \exists s', \exists a,$

$(s2g (\ll a, l, s, s' \gg) = gbis = g_seq \ n \wedge \text{SGPE } (\ll a, l, s, s' \gg) \vee$

$s2g (\ll a, r, s', s \gg) = gbis = g_seq \ n \wedge \text{SGPE } (\ll a, r, s', s \gg)) \wedge$

$s2g \ s = gbis = g_seq \ (n+1)$.

Definition *has_an_escalation* ($g:\text{Game}$) : $\text{Prop} :=$

$\exists g_seq, (\text{has_an_escalation_sequence } g_seq) \wedge (g_seq \ 0 = g)$.

Theorem *Zero_one_game_has_an_escalation*:

$\text{has_an_escalation Alice_Bob nat ge zero_one_game}$.

Theorem *Dollar_game_has_an_escalation*:

$\text{has_an_escalation Alice_Bob nat ge (dollar_game } 0)$.

Index

0, 1 game, 9, 13, 33

Achilles and the tortoise, 10

binary tree, 18

coinduction, 6, 9, 15

dollar auction, 8, 34

escalation, 7, 8, 34

finite game, 21

game, 22

history, 7

infinite word, 9

infiniteness, 6

lazy binary tree, 18

Shubik, 7, 8, 34

strategy profile, 23

Weierstrass, 6, 10

Bibliography

- S. Abramsky and R. Jagadeesan. Games and full completeness for multiplicative linear logic. *Journal of Symbolic Logic*, 59:543–574, 1994.
- P. Aczel. *Non-well-founded Sets*, volume 14 of *CSLI Lecture Notes*. CSLI Publications, Stanford, California, 1988.
- R. Affeldt, M. Tanaka, and N. Marti. Formal proof of provable security by game-playing in a proof assistant. In Willy Susilo, Joseph K. Liu, and Yi Mu, editors, *ProvSec*, volume 4784 of *Lecture Notes in Computer Science*, pages 151–168. Springer, 2007. ISBN 978-3-540-75669-9.
- R. J. Aumann. Backward induction and common knowledge of rationality. *Games and Economic Behavior*, 8:6–19, 1995.
- Y. Bertot. Filters on coinductive streams, an application to Eratostenes’ sieve. In *Typed Lambda Calculus and Applications’05*, volume 3461 of *Lecture Notes in Computer Science*, pages 102–115. Springer-Verlag, 2005.
- Y. Bertot. Affine functions and series with co-inductive real numbers. *Mathematical Structures in Computer Science*, 17(1):37–63, 2007.
- Y. Bertot and P. Castéran. *Interactive Theorem Proving and Program Development Coq’Art: The Calculus of Inductive Constructions*. Springer-Verlag, 2004.
- K. Binmore. Modeling rational players. *Economics and Philosophy*, 3:179–214, 1987.
- K. Binmore. Modeling rational players, Part II. *Economics and Philosophy*, 4: 9–55, 1988.
- J-Ph. Bouchaud. Economics needs a scientific revolution. *Nature*, 455:1181, oct 2008.
- V. Capretta. Common knowledge as a coinductive modality. In Erik Barendsen, Herman Geuvers, Venanzio Capretta, and Milad Niqui, editors, *Reflections on Type Theory, Lambda Calculus, and the Mind*, pages 51–61. ICIS, Faculty of Science, Radboud University Nijmegen, 2007. Essays Dedicated to Henk Barendregt on the Occasion of his 60th Birthday.

- A. M. Colman. *Rational Models of Cognition*, chapter Rationality assumptions of game theory and the backward induction paradox, pages 353–371. Oxford U. Press, 1998. Edited by Mike Oaksford and Nick Chater.
- A. M. Colman. *Game theory and its applications in the social and biological sciences*. London New York : Routledge, 1999. Second edition.
- T. Coquand. Infinite objects in type theory. In Henk Barendregt and Tobias Nipkow, editors, *TYPES*, volume 806 of *Lecture Notes in Computer Science*, pages 62–78. Springer, 1993. ISBN 3-540-58085-9.
- S. Coupet-Grimal. An axiomatization of linear temporal logic in the calculus of inductive constructions. *J Logic Computation*, 13(6):801–813, 2003.
- S. Coupet-Grimal and L. Jakubiec. Certifying circuits in type theory. *Formal Asp. Comput.*, 16(4):352–373, 2004.
- C. Daskalakis, P. W. Golberg, and Ch. H. Papadimitriou. The complexity of computing a Nash equilibrium. *Communications of the ACM*, 52(2):89–97, 2009.
- G. Dowek. *Les métamorphoses du calcul*. Le Pommier, 2007. Grand Prix of Philosophy of the French Academy.
- H. D. Ebbinghaus and J. Flum. *Finite Model Theory*. Springer-Verlag, New York, 1995.
- G. Edgar. *Measure, Topology, and Fractal Geometry*. Springer-Verlag, 2008. 2nd ed.
- R. Fagin. Finite model theory – a personal perspective. *Theoretical Computer Science*, 116:3–31, 1993.
- G. Frege. *From Frege to Gödel*, chapter Concept Script. Harvard Uni. Press., 1967.
- H. Gintis. *Game Theory Evolving: A Problem-Centered Introduction to Modeling Strategic Interaction*. Princeton University Press, 2000.
- J.-Y. Girard. Locus solum: From the rules of logic to the logic of rules. *Mathematical Structures in Comp. Sci.*, 11(3):301–506, 2001. ISSN 0960-1295. doi: <http://dx.doi.org/10.1017/S096012950100336X>.
- J. Y. Halpern. Substantive rationality and backward induction. *Games and Economic Behavior*, 37(2):425–435, 2001.
- J. Harrison. Formal proof – theory and practice. *Notices of the American Mathematical Society*, 55:1395–1406, 2008.

- N. Julien. Certified exact real arithmetic using co-induction in arbitrary integer base. In Jacques Garrigue and Manuel V. Hermenegildo, editors, *FLOPS*, volume 4989 of *Lecture Notes in Computer Science*, pages 48–63. Springer, 2008. ISBN 978-3-540-78968-0.
- H. W. Kuhn, editor. *Classics in Game Theory*. Princeton Uni. Press, 1997.
- H. W. Kuhn. Extensive games and the problem of information. *Contributions to the Theory of Games II*, 1953. Reprinted in Kuhn (1997).
- S. Le Roux. *Abstraction and Formalization in Game Theory*. PhD thesis, École normale supérieure de Lyon (France), January 2008.
- W. Leininger. Escalation and cooperation in conflict situations. *J. of Conflict Resolution*, 33:231–254, 1989.
- P. Lescanne. Deconstruction of infinite extensive games using coinduction. *CoRR*, abs/0904.3528, 2009.
- P. Lescanne. Mechanizing epistemic logic with Coq. *Annals of Mathematics and Artificial Intelligence*, 48:15–43, 2007.
- K. Lorenz and P. Lorenzen. *Dialogische Logik*. Darmstadt, 1978.
- D. Martin. The determinacy of Blackwell games. *The Journal of Symbolic Logic*, 63(4):1565–1581, 1998.
- R. Mazala. Infinite games. In Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors, *Automata, Logics, and Infinite Games*, volume 2500 of *Lecture Notes in Computer Science*, pages 23–42. Springer, 2001. ISBN 3-540-00388-6.
- The Coq development team. *The Coq proof assistant reference manual*. LogiCal Project, 2007. URL <http://coq.inria.fr>. Version 8.1.
- S. Merz. Weak alternating automata in Isabelle/HOL. In Mark Aagaard and John Harrison, editors, *TPHOLs*, volume 1869 of *Lecture Notes in Computer Science*, pages 424–441. Springer, 2000. ISBN 3-540-67863-8.
- R. Milner. *Communication and concurrency*. Prentice-Hall International Series in Computer Science. Prentice Hall, Inc., 1989.
- R. Milner and M. Tofte. Co-induction in relational semantics. *Theor. Comput. Sci.*, 87(1):209–220, 1991.
- D. Mirimanoff. Les antinomies de Russell et de Burali-Forti et le problème fondamental de la théorie des ensembles. *L'Enseignement Mathématique*, 19: 37–52, 1917.
- J. F. Nash, Jr. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences*, 36:48–49, 1950.

- T. Nipkow, L. C. Paulson, and M. Wenzel. *Isabelle/HOL — A Proof Assistant for Higher-Order Logic*, volume 2283 of *LNCS*. Springer, 2002.
- N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, New York, NY, USA, 2007. ISBN 0521872820.
- B. O’Neill. International escalation and the dollar auction. *J. of Conflict Resolution*, 30(33-50), 1986.
- M. J. Osborne. *An Introduction to Game Theory*. Oxford, 2004. ISBN 0195128958.
- M. J. Osborne and A. Rubinstein. *A Course in Game Theory*. The MIT Press, Cambridge, Massachusetts, 1994.
- S. Owre, J. M. Rushby, and N. Shankar. PVS: A prototype verification system. In Deepak Kapur, editor, *11th International Conference on Automated Deduction (CADE)*, volume 607 of *Lecture Notes in Artificial Intelligence*, pages 748–752, Saratoga, NY, jun 1992. Springer-Verlag. URL <http://www.csl.sri.com/papers/cade92-pvs/>.
- D. Park. Concurrency and automata on infinite sequences. In Peter Deussen, editor, *Theoretical Computer Science*, volume 104 of *Lecture Notes in Computer Science*, pages 167–183. Springer, 1981. ISBN 3-540-10576-X.
- R. W. Rosenthal. Games of perfect information, predatory pricing and the chain-store paradox. *Journal of Economy Theory*, 25:92–100, 1981.
- D. Sangiorgi. On the origins of bisimulation and coinduction. *ACM Trans. Program. Lang. Syst.*, 31(4), 2009.
- R. Selten. Spieltheoretische Behandlung eines Oligopolmodells mit Nachfragertrgheit. *Zeitschrift für gesamte Staatswissenschaft*, 121, 1965.
- M. Shubik. The dollar auction game: A paradox in noncooperative behavior and escalation. *Journal of Conflict Resolution*, 15(1):109–111, 1971.
- L. Shustek. Interview Donald Knuth: A life’s work interrupted. *Communications of the ACM*, 51(8):31–37, August 2008.
- R. C. Stalnaker. Belief revision in games: Forward and backward induction. *Math. Soc. SSci.*, 36:31–56, 1998.
- A. Turing. Computability and λ -definissability. *J. Symbolic Logic*, 2:153–163, 1937.
- J. van Benthem. *Logic in Games*. Elsevier, 2006.
- R. Vestergaard. A constructive approach to sequential Nash equilibria. *Inf. Process. Lett.*, 97:46–51, 2006.

-
- R. Vestergaard, P. Lescanne, and H. Ono. Constructive rationality implies backward induction for conscientious players. Research Report JAIST/IS-RR-2006-009, Japan InsT. of Science and Technology, 2006.
- J. von Neumann. Eine axiomatisierung der Mengenlehre. *J. F. Math.*, 154, 1925. Translated as *An axiomatization of Set Theory* in *From Frege to Gödel*, J. van Heijenoort ed., Harvard Univ. Press (1967).
- J. von Neumann. Zur theorie der gesellschaftsspiele. *Mathematische Annalen*, 100:295–300, 1928.
- J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton Univ. Press, Princeton, 1944.
- K. Weierstrass. Über continuirliche Funktionen eines reellen Arguments, die für keinen Werth des letzteren einen bestimmten Differentialquotienten besitzen, 1872. in *Karl Weierstrass Mathematische Werke (1967), Abhandlungen II*. Gelesen in der Königl. Akademie der Wissenschaften am 18 Juli 1872.